AUSTRALIA

PCT/AU2004/001355

Patent Office
Canberra

I, JULIE BILLINGSLEY, TEAM LEADER EXAMINATION SUPPORT AND SALES hereby certify that annexed is a true copy of the Provisional specification in connection with Application No. 2003905438 for a patent by NAPOLEON CORPORATION as filed on 06 October 2003.

COMMONWEALTH OF AUSTRALIA
AUSTRALIA
PATENT OFFICE

WITNESS my hand this
Fourteenth day of October 2004

J. Billingsley

JULIE BILLINGSLEY
TEAM LEADER EXAMINATION
SUPPORT AND SALES

BEST AVAILABLE COPY

P009
Regulation 3.2

AUSTRALIA

Patents Act 1990

PROVISIONAL SPECIFICATION

Invention Title: **A PREMIUM FUNDING SYSTEM**

The invention is described in the following statement:

A PREMIUM FUNDING SYSTEM

The present invention is directed towards the electronic processing of various financial instruments of funding, and in particular to the electronic processing of premium funding services for insurance underwriters, intermediaries and brokers.

BACKGROUND OF THE INVENTION

The present invention seeks to improve fee funding services presently available. Examples such fee funding services include premium funding services for insurance underwriters, intermediaries and brokers. Professional fee funding services for professionals such as doctors, dentists, accountants and lawyers, and membership fee funding services for groups, associations and organisations.

Premium Funding is the term used to describe a widely accepted form of finance which allows consumers to pay insurance premiums via an amortised instalment program, thus enabling the consumer to conserve working capital. Professional Fee Funding is the term used to describe a widely accepted form of finance which allows the clients to pay for professional fees via an amortised instalment program, thus enabling the client to conserve working capital. Membership Fee Funding is the term used to describe a widely accepted form of finance which allows the members of groups, associations and organisations to pay for membership fees via an amortised instalment program, thus enabling the member to conserve working capital.

Should a consumer wish to enter into a premium funding arrangement then security for the loan may be provided by registering a charge on the insurance policy with the underwriter, and also from personal guarantees from the consumer. The charge is released once the loan is paid in full. In the event that the loan is not paid in full, the underwriter is requested to pay the unused portion of the insurance policy to the funder.

Consider the example of a small business with a "general" insurance premium of $5000. In order to assist with cash management, a broker may arrange to amortise this premium over a set period of time, for example 10 months. The broker may add a margin in order to provide the service, the funder then adds an interest component, documentation fees and taxes where applicable and the total amount is then divided into 10 payments. This amount can then be deducted from the consumer's account every month for the next 10 months. The problem with traditional premium funding systems is that they utilise a cumbersome manual reconciliation of settlements for premium funding payments, payments to underwriters and commissions paid to brokers, or all requiring an involved paper trail for all parties.

Utilising traditional premium funding systems, a broker must manually calculate and complete contract paperwork for a client, then forward this paperwork to a head office/funder for approval and processing – all of which can take several days. At this time, the broker has no information regarding the client's credit worthiness or acceptability for such funding. If approved, the head office/funder must then duplicate the client's data and enter the information into their own computer system and arrange for payments to be made, either manually or electronically. Communication between the broker and the head office/funder is difficult and thus extremely limited, with the broker being unaware of what payments and settlements have been performed on this contract.

These systems lack "immediate" credit control and approval systems, with limited "shared knowledge" of the consumer at the time of creating a client, particularly for businesses with multiple offices and/or loan personnel. As a result of this such systems are expensive to manage, incorporating high costs for the generation and organisation of loan documents, administration and management control. Traditional premium funding systems do not allow for flexible payment methods such as supporting split payment facilities over several accounts. For example, a company with multiple divisions may require each division to pay a proportional part of the monthly payment. Traditional premium funding systems are often simplistic, allowing for quotation and document generation only, with little 'full life of loan' application. Traditional systems are also inefficient due to inevitable errors from manual data entry and time spent in processing copious amount of paperwork. Further, traditional systems also do not allow for flexible payment methods and have little integration into existing accounting solutions or third party services such as credit reference checking, debt collection. Traditional contract funding systems utilise either costly, time consuming manual systems, or antiquated computerised systems with limited flexibility. There is therefore a need for an improved system which is more time efficient and relatively easy to use.

OBJECTIVE OF THE INVENTION

An objective of the present invention is to provide a relatively simple system which is more time efficient then the existing manual systems, and provides a certainty of response for both brokers and consumers.

SUMMARY OF THE INVENTION

With the above object in mind the present invention provides in one aspect a funding system including a local processor operable by a broker, said local processor including an input means to allow said broker to input data in respect of a funding request, and said local processor analysing said data using predefined rules to determine whether funding will be offered in response to said funding request; and wherein said local processor synchronises data with a central processor.

In a preferred arrangement said local processor allows said broker to request said predefined rules be overridden for said funding request. Further when said local processor synchronises with said central processor, said central processor performs settlement of said funding request. The local processor and central processor can also perform additional synchronisations allowing the broker to view all settlement activity performed, including any commissions paid, which are associated with the funding request.

BRIEF DESCRIPTION OF THE DRAWINGS

Figure 1 shows the Local Broker Application and the tables which make up the Local Broker Application of the preferred embodiment.

Figure 2 shows the Remote Host Application and the master tables which make up the Remote Host Application of the preferred embodiment.

Figure 3 shows the flow of information between the Local Broker Application and the Remote Host Application and other third party agencies such as financial institutions and credit reference agencies, of the preferred embodiment.

DESCRIPTION OF PREFERRED EMBODIMENT

The present invention can include three main components, namely a Local Broker Application, a Remote Host Application and a Web Interface Application, to enter, record and process client, quotation, contract, payment and settlement transactions.

The Local Broker Application enables the broker operator to enter and record data such as client, quotation, contract, correspondence and payment transaction information (both current and historical) which is ideally stored in individual database tables, according to certain rules and parameters stored on the Local Broker Application. These rules and parameters are ideally created and maintained on a centralised Remote Host Application. The client, quotation or contract created can be initially validated and either accepted or declined by the Local Broker Application, or referred to the Remote Host Application for approval.

If the data has been accepted, or is to be referred, it is forwarded to the Remote Host Application for further processing or replication via a 'synchronisation' methodology. According to certain rules and parameters stored on the Remote Host Application, the data is then validated and authorised by performing velocity error checks on each transaction, and if approved is then processed by the Remote Host Application which can perform the electronic settlement of all contracts and payment schedules between all related parties (client, funder, underwriter, broker, bank). The Remote Host Application is capable of storing a vast number of individual records and databases for a vast number of brokers, and each operator's data can be replicated in full on the Remote Host Application.

The Web Interface Application may be 'cut down' Internet based version of the Local Broker Application which enables brokers to remotely access their own datafiles and create prospects, borrowers, quotations, contracts and messages (SMS, email, and fax) via the Internet from any computer or location via a web browser.

The Remote Host operator can download the GUI (graphical user interface) screens of the Web Interface Application and a subset of the broker's database tables taken from the master database tables stored on the Remote Host Application, onto the web via a HTML page generator allowing the broker to enter, view, select and report on the stored data.

All transactions performed on the Web Interface Application are duplicated and recorded at the Remote Host Application. Upon the next synchronisation between the Remote Host Application and the Local Broker Application, the transactions performed on the Web Interface Application are synchronised to the Local Broker Application in order to update the broker's local datafiles.

The broker enters the borrower's details into the Local Broker Application, along with details of the insurance premium, and selects the terms and fees that will be offered to the borrower.

After selecting the required insurance premium, schedule length e.g. 3 months, 6 months or 12 months, and instalment periods e.g. weekly, fortnightly, monthly, the Local Broker Application will produce a legal and binding contract displaying the insurance premium details, terms and conditions, scheduled payment dates, instalment amounts and any relevant fees. If satisfied, the borrower simply signs a copy of the printed contract.

Upon electronic acknowledgement by the broker that the contracts have been printed and signed by the borrower, the Local Broker Application will then synchronise with the Remote Host Application and forward the details of the borrower and the contract to the Remote Host Application for approval and processing.

Once approved, the Remote Host Application will automatically perform settlement to all stakeholders, deducting the regular payment instalments from the client's preferred bank account on the nominated dates and depositing these directly into the funder's trust account, whilst providing full reporting to all stakeholders.

Should a payment fail, the system will perform automatic re-submissions at a set later date and/or forward delinquent accounts to a nominated debt collection service, notifying all stakeholders of the outcome.

The Invention's unique synchronisation methodology utilises TCP/IP technology and a custom designed 'synchronisation protocol' technology to forward information which has been stored on the Local Broker Application to the Remote Host Application for processing and replication, and vice versa.

TCP/IP (Transmission Control Protocol/Internet Protocol) is the basic communication language or protocol of the Internet. It can also be used as a communications protocol in a private network (either an intranet or an extranet). TCP/IP is a two-layer program. The higher layer, Transmission

Control Protocol, manages the assembling of a message or file into smaller packets that are transmitted over the Internet and received by a TCP layer that reassembles the packets into the original message. The lower layer, Internet Protocol, handles the address part of each packet so that it gets to the right destination.

Application protocols that also use TCP/IP include the World Wide Web's Hypertext Transfer Protocol (HTTP), the File Transfer Protocol (FTP), Telnet (Telnet) which allows a person to logon to remote computers, and the Simple Mail Transfer Protocol (SMTP) which is used when sending emails.

The 'synchronisation protocol' runs in conjunction with TCP/IP technology to enable the Local Broker Application to connect to the Remote Host Application using a telecommunication method such as the Internet or via a VPN. Synchronisation is initiated by the Local Broker Application, with the Remote Host Application detecting any such synchronisation.

When a synchronisation is performed, the Local Broker Application automatically detects any new or modified data created in the application (either entire records or individual fields within a record) and uploads this information to the Remote Host Application.

Conversely, the Remote Host Application also detects any new or modified data created at the Host and downloads this to the Local Broker Application, for example, downloading updates on rules and lending parameters controlled by the Host, modifications to lending rate tables, audit information, stationery templates used in the production of quotations, contracts, and reports, etc.

The synchronisation process is also used to exchange 'messages' between the Local Broker Application and the Remote Host Application, such as broker requests for lending limit increases, finance rate reductions, stamp duty processing and other lending exceptions and the responses that a Remote Host operator may return. This messaging functionality removes the need for physical contact between the broker operator and the Remote Host operator to approve or decline day-to-day processing and business decision requests.

In the preferred arrangement the synchronisation protocol technology uses a series of four letter commands to dictate the type of information which has been sent and what is required to be performed with this information. For example, if the four letter command 'SYNE' is sent by a Local Broker Application to the Remote Host Application, the Remote Host Application may recognise this command as 'the current synchronisation session has ended'.

In addition, as the Remote Host Application is capable of storing a vast number of individual records and databases for a vast number of brokers, each sequence or string of information which is synchronised between a Local Broker Application and the Remote Host Application should contain a 'header' packet of data which can be used by the Remote Host Application to identify, authenticate and validate the credentials of the broker operator who sent it. The header ideally contains the unique identification code of the registered end-user organisation, and the unique 'terminal' identification code of the individual Local Broker Application.

In addition, every piece of system data sent in the sequence or string from the Local Broker Application and attached to the header packet of data should contain a number of database table and record identification codes, allowing the Remote Host Application to replicate the data in the correct master database tables recorded at the Host. For example, if a particular contract is forwarded to the Remote Host Application when a synchronisation is performed, the contract data should contain the identification code of the database table it belongs to [TableID], a unique contract ID [ContractID] and the unique ID of the borrower [BorrowerID] that this contract belongs to.

The present invention has been designed to provide a level of security and functionality as that which is required by major global financial institutions. All data which is passed between the two applications via the Internet or a Virtual Private Network [VPN] is compressed and encrypted using 128 bit key encryption (the global standard for securitised encryption). It will be understood that security encryption can change with acceptable standards.

This method of synchronisation should also support conflict resolution on partially updated records or records which have been simultaneously updated on both the Local Broker Application and the Remote Host Application.

The Local Broker Application can reside on a broker operator's system, allowing the operator to enter, record, process and recall information before forwarding data to the Remote Host Application using the synchronisation process.

The Local Broker Application is available in two configurations; single user (where the broker uses a single application to enter, record, process and recall information) or client server (where the broker may use multiple instances of the application which connect to a local server to enter, record, process and recall information) before forwarding data to the Remote Host application.

The Local Broker Application is comprised of a number of database tables where client, quotation, contract and payment information is stored. Data is entered into these tables by means of GUI [graphical user interface] data entry screens, allowing the broker operator to enter the relevant data into labelled fields, or select data from pre-coded lists, which is then saved, processed and recorded in the applicable database tables. Data is also able to be retrieved or accessed from these tables by means of GUI list screens, allowing the broker operator to search, filter, graph or report the relevant data recorded in the database tables.

The purpose and functionality of these primary database tables is outlined below in further detail. A flow diagram of the Local Broker Application and its primary database tables is shown as Figure 1.

The Broker Database Table stores data related to the identity of the broker using the application such as name, address, contact details, authorised contact people, company ABN numbers, etc. In addition to general contact information, the Broker Database Table can also store the details of the operator's nominated bank accounts for commission payments, settlement payments, etc.

Each Local Broker Application in the preferred arrangement is assigned a Terminal Identification Code and a Broker Identification Code which can be calculated from the registration details of the broker and can be stored in the Broker Database Table. These two identification codes are encoded with every record created on the Local Broker Application and are included with every batch of data forwarded to the Remote Host Application when a synchronisation is performed, allowing the Remote Host Application to identify, authenticate, record and process the data from this terminal (application) and operator.

The Broker Preferences Table stores data related to the nominated defaults and preferences to be used by the broker in the Local Broker Application, such as the default commission percentage, minimum and maximum loan term allowed, minimum and maximum loan value allowed, default documentation fees to be charged, etc.

Whilst some of the defaults and preferences are entered by the operator when initially setting up the system and are forwarded to the Remote Host Application when the operator performs a synchronisation, other defaults and preferences may be set by the Remote Host application and are downloaded to the Local Broker Application and recorded in the Broker Preferences Table when the operator performs the first synchronisation. A number of these defaults and preferences are visible to the operator whilst others are hidden and are used by the application only. When applicable, modifications to the Broker Preferences Table are downloaded during subsequent synchronisations.

The Users Database Table stores data related to each user who has access to the application and their associated system privileges. Every user is assigned a unique User ID and a password, which must be entered and validated when entering the application in order to commence operating the system. The unique User ID and password is a security feature which prohibits unauthorised access to the application or application data.

In addition, each time data is entered into a system and saved, the unique User ID of the user who entered the data recorded with the data. This feature allows data such as clients and quotations to be assigned to, and/or viewed, only by the authorised user. In addition, this information forms the basis of an audit trail which is recorded locally, and on the Remote Host Application, and stores data on the access and modification of records.

Each user can be assigned system privileges for every function and screen in the application, controlling access for viewing and editing data privileges in the application. The primary user of the application is established as the 'Administrator', who retains access to all functions in the application, including the creation and setting of privileges for all other users.

As the present invention is capable of handling and processing contract funding from multiple funders through the one application, the Funders Database Table stores data related to each funder with whom the operator of the Remote Host Application has contract funding arrangements in place, such as the funding type for each funder (e.g. Insurance Premium Funding, Professional Fee Funding, Membership Fee Funding).

The invention allows for three methods of processing on behalf of funders and enables each funder to choose one of these methods for processing;

a) Where the broker performs the initial 'introduction and paperwork' on the Local Broker Application, with the Remote Host Application forwarding an electronic settlement file to the funder, who will then perform its own direct debits and settlements.

b) Where the operator of the Remote Host Application performs an end-to-end solution for a funder by performing all the quotation generation, contract generation, settlement and collection on behalf of the funder.

c) Where the broker has the ability to 'self fund' his own book of business. In this case, the broker becomes the funder and utilises his own pool of funds to fund non-approved contracts, up to a set credit limit. The invention performs an end-to-end solution for the broker/funder by performing all the quotation generation, contract generation, settlement and collection on behalf of the broker/funder.

The Funders Table can be closely linked to the lending rates, lending rules, banking and fees, and forms template sub-set of tables which can contain the funding data and parameters for each funder;

The funding data for each funder is created and maintained by the Remote Host Application. Each individual funder's data is then downloaded to the nominated broker applications with whom the funder has a contract funding arrangement in place, and is recorded in the Funders Table when the operator performs the first synchronisation. When applicable, modifications to the Funders Table are downloaded during subsequent synchronisations.

If the broker has funding arrangements in place with multiple funders, the broker may opt to select from a list of available funders at the time of creating a quotation or contract. Alternatively, according to established rules for funding stored on the Remote Host Application which have been synchronised to the Local Broker Application, the application can automatically choose the appropriate funder according to the parameters of the quotation or contract. For example, a funding rule may be created that assigns all contracts under $25,000 to Funder A and all contracts over $25,000 to Funder B. These rules would override any locally set parameters created at the Local Broker Application and are recorded in the Lending Rules Table.

The Lending Rate Table stores data related to the margins each funder wishes to make on loans provided to borrowers. The Lending Rate Table is used throughout the Local Broker Application to calculate the finance charges on all quotations and contracts.

The Lending Rate Table is created and maintained by the Remote Host Application. The funders' finance rates are downloaded to the Local Broker Application and recorded in the Lending Rate Table when the operator performs the first synchronisation. When applicable, modifications to the Lending Rate Table are downloaded during subsequent synchronisations.

A default margin matrix is applied to every broker upon initial setup on the Remote Host Application. Once configured however, the Lending Rate Table may vary from broker to broker depending upon loan size, loan duration, type of funding (i.e. Premium Funding - for either cancellable or non-cancellable business, Professional Fee Funding, Membership Fee Funding) and the frequency of business.

The Lending Rules Table stores data related to each funder's lending rules or general defaults which apply to this broker when performing funding quotations and contracts. Rules are created as a series of 'IF, THAN, AND/OR, NOT, ELSE' statements and apply to features such as credit checking, use of guarantors, contract value, etc.

For example, a lending rule may be created that assigns all contracts under $25,000 to Funder A and all contracts over $25,000 to Funder B. Each funder has an individual set of rules, for example Funder A may require that only one party must sign the contract at the time of acceptance whereas Funder B may require all parties to sign contracts.

These rules are created and maintained by the Remote Host Application and are downloaded to the Local Broker Application and recorded in the Lending Rules Table when the operator performs the first synchronisation. When applicable, modifications to the Lending Rules Table are downloaded during subsequent synchronisations.

At the time of creating a quotation or contract, the application will automatically perform a cross reference against the Lending Rules table for any general conflict or specific funder conflict. If an appropriate system rule is found e.g. a guarantor is required for the contract, the application will display a system warning or notification to the operator that this must take place before the quotation or contract is concluded.

The Banking and Fees Table stores data related to each funder's bank processing charges and fees which apply to this broker when performing funding quotations and contracts such as transaction fees, rejection fees, quotation, contract, documentation and credit checking fees.

When a funder is initially allocated to a broker, a default set of fees and charges is applied to the broker upon initial setup. Once configured by the Remote Host Application however, the fees and charges may vary from broker to broker depending upon loan size, loan duration, type of funding (i.e. Premium Funding - for either cancellable or non-cancellable business, Professional Fee Funding, Membership Fee Funding) and the frequency of business.

The Banking and Fees Table enables the operator to accurately calculate the cost of processing fees and the profit of a contract, depending on the number of settlements to be made, payments to be processed, rejections for the life of the loan and any credit checks performed.

The Underwriter Database Table is utilised when producing Insurance Premium Funding contracts only and stores data related to insurance underwriters with whom the broker may do business. The contacts, branch locations, classes of insurance (e.g. cancellable & non- cancellable) and premium bank account details (required when performing premium payments) of every insurance underwriter are maintained by the Remote Host Application and are downloaded to the Local Broker Application and recorded in the Underwriter Database Table when the operator performs the first synchronisation. When applicable, modifications to the Underwriter Table are downloaded during subsequent synchronisations.

When creating a Premium Funding contract at the local broker application, the broker must assign one or more insurance policies and underwriters to each contract. The operator selects the required underwriter from a list field (the list field is linked to and displays all underwriters listed in the Underwriter Table) and the branch and class of insurance. The operator also nominates whether the contract is to be settled directly with the broker or the underwriter.

When the contract is finalised and sent to the Remote Host Application for processing, the Remote Host Application identifies the specified underwriter (each underwriter in the Underwriter Database Table is assigned a unique ID code) and automatically settles the amount of the insurance policy premium to the underwriter's premium/trust bank account on the broker's behalf.

When the settlement has been performed, the Remote Host Application automatically notifies the underwriter of the settlement payment, electronically. In the event that the borrower defaults on the contract, the Remote Host Application automatically notifies the underwriter to cancel the policy and requests a refund on the pro-rata balance of the contract.

The Banks Database Table stores data related to banking and financial institutions with whom the operator and borrower may do business. The branch name, address and BSB of every banking and financial institution are maintained by the Remote Host Application. This table is supplied with the first deployment of the application and maintained thereafter with regular updates.

The Banks Database Table is accessed by the Local Broker Application when the broker initially sets up the commission, settlement and trust bank accounts which will be used by the broker and when creating any number of payment methods which may be used by each borrower when making payments. The broker selects the name of the bank or the BSB number from the list field shown on each relevant screen (the list field is linked to and displays all banks listed in the Banks Database Table and may be sorted and searched by each field) and selects the nominated bank. The fields are filled in automatically and the operator then enters the relevant account number.

The Credit Card Table stores data related to each credit card provider with whom the broker/operator has a merchant arrangement such as the card type and default card charges.

Credit card data is created and maintained by the Remote Host Application in a Master Credit Card Table and includes all credit card providers with whom the Remote Host Application is able to perform settlement with.

The default credit cards and card charges are applied to every broker and downloaded to the Local Broker Application and recorded in the Credit Card Table when the operator performs the first synchronisation. When applicable, modifications to the Credit Card Table are downloaded during subsequent synchronisations.

Once downloaded to the Local Broker Application, the broker may individually nominate which credit card providers displayed in the Credit Card Table the broker will accept, for example, the operator may choose to accept Visa and Mastercard but not Amex or Diners.

At the time of creating a contract, if the borrower has nominated to make the scheduled payments with a credit card, the application will cross reference the selected payment method from the Payment Method Table, (i.e. Payment Method = Visa Card), with the Credit Card Table to calculate the operator's merchant fees and card charges for Visa credit card payments. These charges can then be incorporated into the contract value or, at the election of the broker, deducted from the commission paid to the broker or charged to the broker if no commission is to be paid.

The Stamp Duty Table stores data related to the appropriate stamp duty fees and charges for each region. This table is maintained by the Remote Host Application and synchronised with the Local Broker Application. The Stamp Duty Table can also be used for the calculation of any state taxes and VAT duty.

The Stamp Duty Table is accessed by the Local Broker Application when the broker creates a quotation for a borrower. Each broker operates within a default region and stamp duty is calculated on the basis of this region. If required, the default region can be overridden at time of creating a quotation allowing another region to be selected in order to calculate the appropriate duty.

The Forms Template Table stores the templates, per funder, of all stationery used in the production of quotations, contracts, terms and conditions documentation, guarantor forms, etc.

The application is capable of providing funding on behalf of multiple funders, each of whom has individual contract documentation or will require particular information to be collected from the borrower or to be outlined on the contract documentation.

As such, each stationery template recorded in the Forms Template Table is assigned a unique template ID and funder ID, ensuring that the correct contract documentation for each funder is accessed and printed by the system.

The Local Broker Application is able to automatically complete each template with the required data as each stationery template contains a default set of form data which is accessed from a particular database table (the form data and ID of the database table for each template is recorded in the Forms Template Table). For example, a stationery template with the type 'Contract' is assigned to the Contracts Database Field and will retrieve and display the default set of form data for this contract from the Contracts Database Field.

In order to maintain the integrity of all stationery templates, all templates are created and maintained by the Remote Host Application. Any new or modified templates are downloaded to the Local Broker Application and replicated in the Forms Template Table on the Local Broker Application when a synchronisation is next performed. If required, an authorised representative of the funder may 'brand' or customise form templates with the broker's logo.

The Report Template Table stores the templates of all reports used and/or created in the system. The Local Broker Application is initially deployed with a number of system reports which are created and maintained by the Remote Host Application. Any new or modified templates are downloaded to the Local Broker Application and replicated in the Report Template Table on the Local Broker Application when a synchronisation is next performed.

In addition, the broker may create and edit new reports according to the broker's own requirements, which are assigned a name and unique ID and are also recorded in the Report Template Table. Any report created and saved by the broker appears in the list of available reports to select from.

The Local Broker Application is able to automatically produce each report with the required data as each report template contains a default set of form data which is accessed from a particular database table (the form data and ID of the database table for each template is recorded in the Report Template Table). For example, a report template with the type 'Borrowers' is assigned to the Borrowers Table and will retrieve and display the required form data for this report from the Borrowers Table. Additional data may also be accessed from other tables in the application.

The Borrower Database Table stores data related to every client with whom the operator does business such as contact and address details, employment details, identification details, credit limit, status, etc. Depending upon the customer type (e.g. individual, company, trust, partnership, etc.) the data required for entry into the application may vary.

Each client is assigned a unique Borrower Identification Code and these codes are stored in the Borrower Database Table. The Borrower Identification Code is recorded with every note, quotation, contract or payment assigned to the borrower, and should be cross referenced in other Quotation and Contract database tables, allowing the operator to search, filter and display data such as all quotations for a particular borrower, all contracts for a particular borrower, etc.

When a new prospect/borrower is created by an operator, the application performs a number of velocity error checks on the Borrower Database Table for items such as same name, address and contact details as that of another borrower, before declining or accepting the new borrower record.

Every new client entered into the system is initially assigned the status flag of 'Prospect'. When a quotation for funding is accepted by the client, the client's status flag is converted to 'Customer'. Alternatively, the operator may manually convert the prospect's status flag to Customer if desired.

The operator may enter multiple addresses (e.g. shipping, postal, home, business, etc.) for each borrower and multiple contact details for phone, fax, mobile and email addresses. The borrower can also nominate a primary method of contact to be used when receiving messages i.e. via SMS or email or letter.

In addition, multiple 'contacts' can be assigned to each borrower. For example, if the borrower is a business or company, any number of contact people at the business and their individual contact details can be assigned under this borrower.

For marketing purposes, the application also enables the operator to assign borrowers to specific industries or groups (these categories are created and maintained by the operator on the Local Broker Application). Borrowers can also be assigned any number of operator-defined keywords or marketing labels, allowing the operator to search and filter borrowers based upon these keywords.

Borrowers can also be assigned a 'pre-approved' credit limit. This limit is established by the operator of the Remote Host Application and will allow a quotation to take place outside of the normal lending ranges applied to a broker. For example, if a pre-approved credit limit of '0' is recorded in the Borrower's Table for a particular borrower, the application will ignore the field and use the standard quotation limit ranges when creating a quotation. If a value is entered into the pre-approved credit limit field, the application will consider this value when creating a quotation. A factor that may affect the pre-approved credit limit is the current balance (stored in the broker application and updated at the time of synchronisation) of the borrower's contracts, as the borrower may have existing contracts which will then exceed the pre-approved credit limit.

The Borrower Table also records the details of the borrower's previously stamped stamp duty payments. If the borrower has existing contracts with the broker and an existing funder, the amount of stamp duty already paid to the funder is recorded in the Borrower Table and is automatically displayed on the quotation screen and taken into account when calculating stamp duty for the contract.

If the broker is creating a quotation on behalf of a borrower using a particular funder and the borrower claims they have previously paid stamp duty to this funder in a previous loan transaction generated by a non-related party elsewhere, the broker may perform a Stamp Duty Adjustment Request.

Note, as this procedure must be authorised by the borrower, the application will automatically generate the appropriate form which must be signed by the borrower. The application will also request acknowledgement from the broker that the authorisation form has been signed and executed correctly. The

Stamp Duty Adjustment Request is then synchronised to the Remote Host Application, along with the details of the previous stamp duty payment.

The Remote Host operator will make enquiries with the funder as to when the stamp duty was paid and the amount paid. If verified, the Remote Host operator will insert the details of the previous stamp duty payment into the borrower's record on the Master Borrower Table held at the Host and download this information to the broker, updating the Local Broker Application with the borrower's new stamp duty data. The Remote Host Application will forward an electronic message to the broker, notifying that the Stamp Duty Adjustment Request has been approved or rejected.

If Membership Fee Funding is to be performed using the application, borrowers can be assigned a membership number. A unique 'starting' membership number is created by the Remote Host Application, with the ability to select some or all of the broker's borrowers in order to assign them with a membership number. This 'default' membership number will be automatically displayed when entering a borrower's membership screen, however the broker can override this number and manually allocate a membership number to the borrower.

Borrowers can also be assigned a membership credit limit and membership 'start' and 'expiry' dates, allowing the application to display whether the borrower is a current or expired member.

When a synchronisation is performed between the Local Broker Application and the Remote Host Application, the Local Broker Application automatically detects any new or modified data in the Borrower Database Table and forwards this information (along with the relevant Borrower Identification Codes) to the Remote Host application, where it is replicated in the Master Borrower Database Table stored at the Host.

The Borrower Notes Table stores the details of every note created in the application which has been assigned to the borrower (identified by the unique Borrower Identification Code and Note Identification Code which are recorded with each note).

Borrower notes can be either 'system notes' which are automatically created by the application (a system entry such as the creation date of a contract) or 'operator notes' which are text notes applicable to the borrower and have been manually entered by the operator (e.g. any pertinent notes related to a client's contract). Operator notes for a particular borrower can be entered into the application via the individual Borrower, Quotation and Contract screens.

When creating an operator note, brokers have the ability to perform or assign 'actions' to the note which varies the data displayed and stored with the note, such as Appointment, Sales Lead, Phonecall, etc. For example, if creating a note about an appointment with a client, the broker may select the description 'Appointment' from an operator-defined list of action types. The GUI note screen will then allow the broker to select the date and time of the appointment from a system calendar and enter any information about the relevant appointment.

Alternatively, the broker may create a note type of 'Phonecall'. In this case, the GUI note screen will display a list of contact people associated with this borrower and their telephone numbers. The broker selects the appropriate contact person and enters information regarding the telephone conversation. The broker may also assign a status to the phonecall note, such as 'Borrower Busy', 'Left Message' or 'No Answer'.

The broker also has the ability to utilise a 'Timer' function in conjunction with each note and action. For example, if creating a note with an action type of Phonecall, the broker can click on the Timer icon displayed on the Action - Phonecall GUI screen to start the Timer to record the length of the phonecall and assign this time to the note. The Timer can also be used to time appointments, meetings or time spent working on a borrower's contract for billing out purposes.

When a synchronisation is performed between the Local Broker Application and the Remote Host Application, the Local Broker Application automatically detects any new or modified data in the Borrower Notes/Messages Database Table and forwards this information (along with the relevant Borrower Identification Codes) to the Remote Host application, where it is replicated in the Master Borrower Notes/Messages Table stored at the Host.

In the case of 'system notes' created by the Remote Host application, these are downloaded to the Local Broker Application and recorded in the Borrower Notes Table when the next synchronisation is performed.

The Borrower Messages Database Table stores the details of every message created or received by the application which has been assigned to the borrower (identified by the unique Borrower Identification Code and Message Identification Code which are recorded with each message).

The Payment Methods Table stores the details of any number of payment methods which have been assigned to each borrower, such as account name, BSB number, account number, credit card number, etc. Payment methods are the bank account or credit card details with which a borrower chooses to make contract payments.

Borrower payment methods can be created by the operator when entering a new client, or on the fly at the time of creating a contract. For example when creating a new payment method, the operator can select either;

a) The bank name and BSB from the list field on each relevant screen (the list field is linked to the Banks Database Table and displays all banks listed in the Banks Database Table) and inserts the address and BSB number of the nominated bank. The operator then enters the relevant account name and number.

b) The credit card type from the list field on each relevant screen (the list field is linked to the Credit Card Database Table and displays all credit card types accepted by the operator which are listed in the Credit Card Database Table). The operator then enters the name of the credit card holder, expiry date and any additional security information e.g. Visa security number located on the back of the card.

When creating a contract, the borrower must nominate a payment method which will be used to make the scheduled payments. If the borrower already has payment methods created the operator selects the required payment method from the list field (the list field is linked to the Payment Methods Table and displays all payment methods which have been assigned to this borrower, using the borrower's unique ID code, and which are recorded in the Payment Methods Table).

Borrowers may also nominate to perform a 'split payment' on one or more scheduled payments by assigning a percentage or set dollar figure of each payment to individual payment methods. For example, a business client may have a monthly scheduled payment of $2,500 per month. The business wishes to 'split' the monthly scheduled payments between two of its branch localities with Branch A paying 50% of the payment from Bank Account A and Branch B paying 50% of the payment from Bank Account B. By creating two different payment methods (one for Bank Account A and one for Bank Account B) the operator can split the monthly scheduled payments between the two. Each monthly scheduled payment would then comprise a debit of $1,250 from Bank Account A and another debit of $1,250 from Bank Account B.

The application also allows the borrower to change the nominated payment method they are using to make scheduled payments, or to pay one or more scheduled payments with a different payment method.

When a synchronisation is performed between the Local Broker Application and the Remote Host Application, the Local Broker Application automatically detects any new or modified data in the Payment Methods Database Table and forwards this information (along with the relevant Borrower Identification Codes) to the Remote Host application, where it is replicated in the Master Payment Methods Table stored at the Host.

The Messages Database Table stores the details of every message created in the application which has been sent to or received from a borrower (identified by the unique Borrower Identification Code and Message Identification Code which is recorded with each message).

The invention enables brokers to create and send messages from the Local Broker Application directly to their client base, via a number of methods such as single and Bulk SMS Messages, Single and Bulk Emails, Single and Bulk Faxes, and, Single and Bulk Letters.

The Messages Table on the Local Broker Application stores the data for each message such as Message ID, Borrower ID, User ID of the operator who created it, message type (SMS, email, fax, letter) date and time sent/to be sent and message status (either Pending, Sent, Rejected or Received).

The Messages Table is also represented by a GUI list screen, allowing the broker to view, search and recall each message recorded in the database tables.

Messages are created in the system by means of GUI data entry screens, allowing the broker to enter the text of the message into a number of individual message creation screens e.g. Create New SMS Message screen.

Brokers may create single messages or a group of bulk messages which can be sent to a range of borrowers by 'filtering' the contact and marketing information stored for each borrower. Messages can also be created quickly by using pre-set message templates. For example, in the case of a standard email sent to customers every month, the broker simply selects a standard email template and the text is automatically inserted by the application.

The broker may also insert a range of merge fields from the database tables into the text of messages in order to personalise messages. Merge fields are available for every field in every database table e.g. the Borrower Table and the Contracts Table. For example, by inserting the merge field <FIRST NAME> into an email message, the application will automatically insert the borrower's first name.

Messages can also be automatically created and sent by utilising message triggers which are linked to key dates. Message triggers facilitate event driven message creation, for example, a message trigger can be used to automatically create and send a contract expiry message to the borrower on a particular date. Brokers can create single or bulk message triggers for borrowers.

When sending SMS, email and fax messages, the Local Broker Application synchronises with the Remote Host Application and forwards the messages to the Host for processing via the Remote Host Application's SMS, email and fax servers. Each message is replicated in full in the Master Messages Table stored on the Remote Host Application. Any messages which are unable to be sent or are returned as unsent are assigned a status of 'Rejected' and are downloaded to the Messages Table on the Local Broker Application when the next synchronisation is performed.

The Quotations Database Table stores the details of every quotation created in the system and assigned to a borrower (using the unique Borrower Identification Code), such as borrower details, loan start date, number of instalments, draw down dates and amounts, commission, stamp duty, fees and charges and an associated payment schedule for the quotation.

Every contract must first be created as a quotation. When creating a quotation, the application will prompt the operator to select the type of funding quotation required e.g. Insurance Premium Funding, Professional Fee Funding or Membership Fee Funding. Note, the quotation and contract screens should ideally differ slightly for each funding type. The appropriate quotation screen for the funding type required will then be displayed.

When creating a quotation, the operator enters the details of the borrower (an existing borrower can be selected from a list field which is linked to the Borrower Table or the operator can create a new prospect at the time of creating the quotation.

The operator then enters the term of the loan and when the associated draw downs are to take place. If creating an Insurance Premium Funding quotation, the draw downs can be separated into cancellable and non-cancellable amounts. If creating a Professional Fee Funding quotation or Membership Funding quotation, the operator can enter a description of the fees and services offered under this quotation or select the fees and services from a pre-defined list.

The application then cross references the Preferences Table (to determine the default commission, stamp duty and documentation fee amounts) and the Lending Rates Table (to determine the finance rates for the loan amount) and calculates the total amount of the quotation.

A deposit may or may not be mandatory for each quotation and subsequent contract. This requirement is controlled by the Remote Host Application and forms part of the Lending Rules, varying from broker to broker and funder to funder. The deposit may be either a percentage of the overall contract, a dollar value or set amount. If the deposit is paid directly to the broker by either cash or cheque, the application will deduct the deposit amount from the balance of the contract to be funded. If the deposit is to be paid via DDR, this value will be included with the funding contract and collected by the application as part of the first instalment.

The application will also automatically calculate and display other data associated with the quotation such as the broker's overall 'exposure' for this quotation for example, if creating an Insurance Premium Funding quotation for cancellable premium funding, the Exposure screen will display the refund amount the broker can expect to receive from the underwriter should the borrower default on payment of the contract. The application will also calculate and display data associated with the 'cost benefits' to the broker of this particular quotation, based upon the finance rate, commission, etc. In addition, the broker can manually enter figures into the cost benefits screen in order to view the cost benefits of lowering or raising the finance rate, etc.

From within the quotation screen, brokers can also create and view operator notes which are assigned to this borrower and this quotation in particular. The note is assigned a unique note ID and is recorded in the Borrower Notes Database Table along with the ID of the borrower and the ID of this quotation.

Prior to saving and accepting the quotation, the application can perform a series of velocity checks, for example, the application can cross reference the amount of the loan against the Minimum and Maximum Loan Value fields in the Preferences Database Table to see if the loan amount fits the default criteria for this broker. The application can also cross reference the Date of Birth field in the Borrower's Table against the Minimum Borrower Age field in the Preferences Database Table to ensure the borrower fits the criteria. If an invalid entry is found, the application will notify the broker that the quotation cannot be accepted.

Should the broker wish to print a copy of the quotation for the borrower, the application retrieves the appropriate quotation template from the Forms Template Table and prints out a copy of the quotation with the relevant details inserted.

Should the borrower accept the quotation terms and wish to convert the quotation to a contract, a copy of the original quotation remains in the Quotations Database Table and a new contract containing all the relevant data from this quotation is created in the Contracts Database Table and assigned a new Contract Identification Number.

Prior to converting a quotation to a contract, the application will perform a velocity check against the current date and the original date of the quotation, and cross reference this with the Quotation Expiry Date field recorded in the Preferences Database Table. If the quotation has expired, a warning message will be displayed to the broker. If desired, the broker may then copy the expired quotation to create a new quotation.

When a synchronisation is performed between the Local Broker Application and the Remote Host Application, the Local Broker Application automatically detects any new or modified data in the Quotations Database Table and forwards this information (along with the relevant Borrower Identification Codes, etc.) to the Remote Host application, where it is replicated in the Master Quotations Table stored at the Host.

The Contracts Database Table stores the details of every contract created in the system and assigned to a borrower (using the unique Borrower Identification Code), such as borrower details, loan start date, number of instalments, associated invoice, membership or insurance policy details, commission, stamp duty, fees and charges, an associated payment schedule for the contract and a list of settlements performed during the contract. Every contract must first be created as a quotation and if accepted, is then converted to a contract.

If creating an Insurance Premium Funding contract, the broker must assign an insurance policy or policies to the contract. The operator selects the required underwriter from a list field (the list field is linked to and displays all underwriters listed in the Underwriter Table) then enters the policy number and details of the policy.

The system will then cross reference the Lending Rules table to ensure that the ratio of cancellable and non-cancellable premiums falls within the parameters configured by the Remote Host Application e.g. a lending rule may exist where non-cancellable insurance may only account for a maximum of 50% of the contract value. If approved, the new insurance policy/policies will be recorded in the Insurance Policy Table.

If creating a Professional Fee Funding or Membership Fee Funding contract, the broker must assign an invoice for the professional fees or good and services to be funded to the contract. The broker enters the date, amount and invoice number of the relevant invoice. The broker then selects the Invoice type from a picklist e.g. professional fees, goods and services, both, etc. The data in this picklist is controlled by the Remote Host Application. The broker then enters a description of the professional fees or goods and services covered by the invoice. This information can be selected from a picklist which the broker maintains locally.

The system will then cross reference the date of the invoice with the 'Maximum Invoice Age' field configured by the Remote Host in the Lending Rules table to see if the invoice falls within the correct funding guidelines, e.g. the invoice must be less than 90 days old. If the invoice date exceeds this field, the application will not allow the funding to take place. The system will also cross reference the Lending Rules table to ensure that the ratio of professional fees to goods and services, and vice versa, falls within the parameters configured by the Remote Host Application e.g. a lending rule may exist where goods and services may only account for a maximum of 50% of the contract value. If approved, the new invoice will be recorded in the Invoice Table.

When creating a contract, the borrower must nominate a payment method which will be used to make the scheduled payments. If the borrower already has payment methods created the broker selects the required payment method from the list field (the list field is linked to the Payment Methods Table and displays all payment methods which have been assigned to this borrower, using the borrower's unique ID code, and which are recorded in the Payment Methods Table). If no payment method has been created for this borrower, the operator must create a new payment method by either;

a) Selecting the bank name and BSB from the list field on each relevant screen (the list field is linked to the Banks Database Table and displays all banks listed in the Banks Database Table) and inserts the address and BSB number of the nominated bank. The operator then enters the relevant account name and number.

b) Selecting the credit card type from the list field on each relevant screen (the list field is linked to the Credit Card Database Table and displays all credit card types accepted by the operator which are listed in the Credit Card Database Table). The operator then enters the name of the credit card holder, expiry date and any additional security information e.g. Visa security number located on the back of the card.

When saved, the new payment method will be recorded in the Payment Methods Table for this borrower.

Borrowers may also nominate to perform a 'split payment' on one or more scheduled payments by assigning a percentage or set dollar figure of each payment to individual payment methods. For example, a business client may have a monthly scheduled payment of $2,500 per month. The business wishes to 'split' the monthly scheduled payments between two of its branch localities with Branch A paying 50% of the payment from Bank Account A and Branch B paying 50% of the payment from Bank Account B. By creating two different payment methods (one for Bank Account A and one for Bank Account B) the operator can split the monthly scheduled payments between the two. Each monthly scheduled payment would then comprise of a debit of $1,250 from Bank Account A and another debit of $1,250 from Bank Account B.

The application also allows for 'third party' settlement of contracts. The ability for a broker to perform a third party settlement is configured by the Remote Host Application, and in each case, the borrower must have a valid and current membership number. If a third party settlement is to be performed, the broker selects the third party broker from a list of third party brokers which is configured by the Remote Host Application. The contact and bank account details of the third party broker will then be displayed. The broker then has the option to send the contract to the borrower and third party broker via email, fax or letter for signing and return. Prior to synchronising the contract with the Remote Host Application, the Local Broker Application will prompt the broker to electronically acknowledge that the borrower and the third party broker have viewed and signed all the appropriate contract documentation such as the contract itself, the terms and conditions and credit checking authorisation forms.

From within the contract screen, brokers can also create and view operator notes which are assigned to this borrower and this contract in particular. The note is assigned a unique note ID and is recorded in the Borrower Notes Database Table along with the ID of the borrower and the ID of this contract.

Prior to saving and accepting the contract for processing, the application will perform a series of velocity checks against the borrower. For example, the application will cross reference the Contracts Database Table to see if the borrower has existing contracts with the broker. If existing contracts are found, the application will cross reference the Credit Limit field in the Borrowers

Database Table to check if the new contract and balance of the existing contracts exceeds the borrower's credit limit.

If the contract is accepted by the Local Broker Application and is to be sent to the Remote Host Application for processing, the application will create new entries in the Scheduled Payments Table with the dates, amounts and nominated payment method for this contract.

The system will also print a number of mandatory contract documents by retrieving the appropriate contract templates from the Forms Template Table such as contract form, terms and conditions form, guarantor's form, etc and print out a copy of the documentation with the relevant details inserted for signing by the borrower.

Prior to synchronising the contract with the Remote Host Application, the Local Broker Application will prompt the broker to electronically acknowledge that the borrower has viewed and signed all the appropriate contract documentation such as the contract itself, the terms and conditions and credit checking authorisation forms.

The application will display a number of Yes/No questions such as 'Has the contract been printed?' and 'Has the contract been signed'. If the broker answers 'No' to any of the questions, the contract is saved and recorded by the Local Broker Application but not synchronised with the Remote Host Application for processing. For example, if the contract has been printed but not yet signed by the borrower, the contract remains stored on the Local Broker Application until the broker acknowledges that the contract has been signed. The user ID and date and time of the broker's acknowledgement is recorded by the application in the Audit Trail Table.

If the contract is not signed and executed with the default expiry period set up in the broker's Preferences, the application will automatically detect and notify the broker that the contract has expired and therefore the finance rates used may no longer be accurate. In this case, the broker can copy the original quotation in order to create a new quotation with the latest finance rates and convert this quotation to a contract for signing.

Once a contract has been signed and acknowledged by the broker as executed, a synchronisation is then performed between the Local Broker

Application and the Remote Host Application, forwarding the new or modified contract data (along with the relevant Borrower Identification Codes, etc.) to the Remote Host application, where further velocity checks may take place.

If the contract is accepted by the Remote Host Application after the additional velocity checking takes place a copy of the contract is replicated in the Master Contracts Table stored at the Host. In addition, new entries detailing the contract ID code, borrower ID code, dates, amounts and nominated payment method ID code will be created in the Master Scheduled Payments Table stored at the Host.

Alternatively, at the time of saving a contract, the broker may opt to perform a 'real-time velocity check' of the contract. In this instance the broker must be 'online' (connected to the internet), allowing the Local Broker Application to connect to the Remote Host Application and perform all the necessary velocity error checks on the contract immediately. If approved, the contract moves from a pre-approved state to an approved state. The ability to perform 'real time velocity checking' will be based on rules configured by the Remote Host Application and will be funder dependent, varying from funder to funder and broker to broker. It may also require the system to perform a real-time check with the credit reference agency and thus may take some time.

The Guarantors Database Table stores data related to every guarantor associated with a borrower such as contact and address details, identification details, credit information, etc.

At the time of creating a contract, according to the individual lending rules of each funder (recorded in the Lending Rules Table), it may occur that a borrower's contract may only be approved if a suitable guarantor accepts responsibility for the contract.

If the borrower already has guarantors linked with their records, the operator selects the required guarantor from the list field on the relevant borrower/contract screen (the list field is linked to the Guarantor Table and displays all guarantors which have been linked to this borrower and which are recorded in Guarantors Table). If no guarantor has been created for this borrower, the operator must create a new guarantor via the borrower's screen. When saved, this guarantor's details will be recorded in the Guarantor Table.

At the time of printing a contract, if guarantor forms are also required to be printed for signing, the application will automatically retrieve the nominated guarantor's data from the Guarantor Table and insert this into the guarantor documentation.

Prior to synchronising the contract with the Remote Host Application, the Local Broker Application will prompt the broker to electronically acknowledge that the guarantor has viewed and signed all the appropriate guarantor documentation associated with the contract.

The Insurance Policy Database Table is utilised when producing Insurance Premium Funding contracts only and stores the details of every insurance policy attached to a contract in the system and assigned to a borrower (using the unique Borrower Identification Code), such as underwriter, policy number, policy amount, type of policy, etc.

When creating a contract, the broker must assign an insurance policy or policies to the contract. The broker selects the required underwriter from a list field (the list field is linked to and displays all underwriters listed in the Underwriter Table) then enters the policy number and details of the policy. The broker may also nominate for the underwriter to receive settlement for the policy when the first scheduled payment is performed, or for the settlement amount to be paid to the broker on the underwriter's behalf.

The system will then cross reference the Lending Rules table to ensure that the ratio of cancellable and non-cancellable premiums falls within the parameters configured by the Remote Host Application e.g. a lending rule may exist where non-cancellable insurance may only account for a maximum of 50% of the contract value. If approved, the new insurance policy/policies will be recorded in the Insurance Policy Table.

When a synchronisation is performed between the Local Broker Application and the Remote Host Application, the Local Broker Application automatically detects any new or modified data in the Insurance Policy Table and forwards this information (along with the relevant Borrower and Contract Identification Codes) to the Remote Host application, where it is replicated in the Master Insurance Policy Table stored at the Host.

The Invoice Database Table is utilised when producing Professional Fee Funding or Membership Fee Funding contracts only and stores the details of every third party invoice attached to a contract in the system and assigned to a borrower (using the unique Borrower Identification Code), such as invoice date, invoice number, description, etc.

When creating a Professional Fees contract or a Membership Fees contract, the broker must assign an invoice for the professional fees or good and services to be funded to the contract. The broker enters the date, amount and invoice number of the relevant invoice. The broker then selects the Invoice type from a picklist e.g. professional fees, goods and services, both, etc. The data in this picklist is controlled by the Remote Host Application. The broker then enters a description of the professional fees or goods and services covered by the invoice. This information can be selected from a picklist which the broker maintains locally.

The system will then cross reference the date of the invoice with the 'Maximum Invoice Age' field configured by the Remote Host in the Lending Rules table to see if the invoice falls within the correct funding guidelines, e.g. the invoice must be less than 90 days old. If the invoice date exceeds this field, the application will not allow the funding to take place. The system will also cross reference the Lending Rules table to ensure that the ratio of professional fees to goods and services, and vice versa, falls within the parameters configured by the Remote Host Application e.g. a lending rule may exist where goods and services may only account for a maximum of 50% of the contract value. If approved, the new invoice will be recorded in the Invoice Table.

When a synchronisation is performed between the Local Broker Application and the Remote Host Application, the Local Broker Application automatically detects any new or modified data in the Invoice Table and forwards this information (along with the relevant Borrower and Contract Identification Codes) to the Remote Host application, where it is replicated in the Master Invoice Table stored at the Host.

The Scheduled Payments Table stores the details of every scheduled payment attached to every contract in the system, such as date of instalment, amount of instalment, etc.

When a quotation is created, the application automatically calculates a schedule of payments for the quotation based upon the dates of the draw downs, the amount of the loan, the finance rate to be applied to the loan and any fees or charges which are associated with funding the loan. The schedule of payments is also based on default parameters listed in the Preferences Database Table such as the Minimum and Maximum Loan Term.

For example, a borrower requires funding for an insurance premium of $25,000. This premium attracts a finance rate of 6.929% (or $1,182.28) plus other documentation fees and charges totalling $712.50. The entire amount to be funded is calculated at $26,894.78. The date of the first draw down is entered as 10/01/2003. The application cross references the Maximum Loan Term field in the Preferences Database Table which has been defaulted to 12 months and then calculates the Scheduled Payments for this contract by dividing the amount to be funded of $26,894.78 into 12 equal monthly payments of $2,241.24, commencing on the 10/01/2003 and finishing on the 10/01/2004. (Note, this calculation takes into consideration that the broker has nominated for his commission to be paid evenly over the contract term and not in full with the first scheduled payment).

If the quotation is accepted and converted to a contract, the scheduled payments are recorded in the Scheduled Payments Table (along with the ID code of the nominated payment method for this contract) and are outlined in full on the printed contract documentation. The contract is then synchronised with the Remote Host application, and if accepted, the Scheduled Payment Table for this contract is replicated in the Master Scheduled Payments Table stored at the Remote Host and used to perform settlement of contract payments.

In the case that a particular scheduled payment is returned as rejected, based upon parameters defined at the Remote Host Application such as the number of times the payment has been rejected, the Host operator will either choose to re-schedule the payment (a suggested re-scheduled payment date will be provided by the Host but can be overridden by the Remote Host operator if not acceptable) in which case the Remote Host Application will insert a 're-scheduled payment' in the Master Scheduled Payments Table upon the rules and parameters defined in the Rejection Log Table such as the default date for the next payment attempt (e.g. in 7 days time) and the amount of any additional fees or charges setup in the Preferences Database Table for rejected scheduled payments e.g. $30 for each missed payment. The re-scheduled payment is downloaded to the broker's Schedule Payments Table when a synchronisation is next performed. Alternatively, the Host operator may choose to issue the borrower with a final notice or cancel the contract. The application will then automatically notify the broker and borrower with the appropriate notifications.

The Settlement Database Table stores the details of the settlement of every payment performed by the application such as settlement status (e.g. pending, paid, etc.) contract ID, payment date, payment amount, bank account details, payee, etc.

This information is sourced and maintained by the Remote Host Application and is downloaded to the Local Broker Application when a synchronisation is performed and replicated in the Settlement Database Table on the Local Broker Application.

The Settlement Database Table enables the operator to open any contract required and view a list of all settlements (current and historical) assigned to this particular contract or to view a master GUI list of all settlements (current and historical) for all contracts in the application, with the ability to search or filter this information to find a particular settlement or range of settlements.

The Rejection Log Table stores the details of every scheduled payment which has been returned as rejected by the Remote Host Application, such as the borrower ID code, contract ID code, bank transfer ID, retry days, number of rejections for this payment etc.

This information is sourced and maintained by the Remote Host Application and is downloaded to the Local Broker Application when a synchronisation is performed and replicated in the Rejection Log Table on the Local Broker Application.

The Rejection Log Table is a hidden table which is not accessible by the operator, but is used by the application for cross referencing and processing purposes. For example, if a particular scheduled payment is returned as rejected, a record is created in the Rejection Log Table and the application will create a 're-scheduled payment' in the Scheduled Payments Table upon the rules and parameters defined in the Rejection Log Table such as the default date for the next payment attempt (e.g. in 7 days time) and the amount of any additional fees or charges setup in the Preferences Database Table for rejected scheduled payments e.g. $30 for each missed payment. The modified data is uploaded to the Master Scheduled Payments Table stored at the Remote Host when a synchronisation is next performed.

The Rejection Log Table keeps a record of the number of rejection attempts for each scheduled payment (this information is also cross referenced in the Scheduled Payments Table) and after a predefined number of attempts which can be stored in the Preferences Database Table, will forward the details of the outstanding amount to the operator or nominated debt collector for payment.

The Audit Trail Table tracks the creation and progress of every quotation, contract and borrower in the system, recording the user name, times and dates of every function associated with each record. For example, when printing a quotation or answering each Yes/No question when finalising a contract, the user's name, date and time are recorded in the Audit Trail Table. If these functions are performed more than once, each occasion is recorded.

The Audit Trail Table is a hidden table which is not accessible by the broker. When a synchronisation is performed between the Local Broker Application and the Remote Host Application, the data from the Audit Trail Table on the Local Broker Application is uploaded and recorded in the Master Audit Trail Table stored on the Remote Host Application.

The Last Synchronisation Table stores a summary of the details of the very last synchronisation performed between the Local Broker Application and the Remote Host Application such as the date and time the synchronisation was performed.

This information is automatically recorded in the Last Synchronisation Table by the Local Broker Application and in the Master Last Synchronisation Table on the Remote Host Application when a synchronisation is performed.

The Last Synchronisation Table is a hidden table which is not accessible by the broker, but is used by the application for cross referencing and backup purposes. For example, should the data on a Local Broker Application become corrupted, the broker may either;

a) If the broker has a recent backup copy of the data restore the latest backup. If work has been performed and synchronised on the application since the backup was made, the broker can request a copy of the data from the very last synchronisation (this information is recorded in the Last Synchronisation Table).

b) If the broker does not have a recent backup copy of the data, the broker can request a 'Restore of All Data from the Host' which will include all the broker's information which has been replicated at the Host, up to the very last synchronisation performed.

The Synchronisation Conflicts Table stores the details of any errors, conflicts or failed synchronisations which have occurred during past synchronisations, such as the date and time the synchronisation conflicts occurred and the database table and fields which were affected.

This information is automatically created and maintained by the Remote Host Application. Should a synchronisation conflict occur, the data is downloaded from the Master Synchronisation Error Table to the Local Broker Application and replicated in the Synchronisation Conflicts Table.

If a problem is severe then the system should prompt the broker to contact the Remote Host operator for assistance. The Remote Host operator will instruct the broker on any processes which need to be performed in order to correct the conflict.

The Local Broker Application incorporates a functionality called 'Task Manager'. The Task Manager is an automated 'To Do' list and notifies the broker of tasks which are due, yet to be completed or outstanding. The Task Manager is not linked to a particular database table, but instead obtains information from many database tables and converts the records from these tables into a GUI list screen, allowing the broker to view the outstanding tasks and take action.

For example, on a daily basis the Task Manager functionality sweeps the database tables and locates all quotations and contracts which are awaiting completion and displays these in the Task Manager GUI list, along with a visual display of the status and outstanding task for each quotation and contract e.g. Contract Printed, DDR Signed, Awaiting Host Response, etc. The Task Manager functionality also locates any prospects which are due to be contacted, or borrowers who have quotations or contracts which are about to expire. In addition, any messages which are due to be sent on this day are also located and displayed in the Task Manager GUI list screen.

The Remote Host Application can reside on a remote computer server/s and performs the 'validation' and 'replication' of broker data and the 'electronic settlement' of all payment transactions. These three functions are outlined in further detail below.

As each broker's data is replicated in full on the Remote Host application, and the Remote Host application is capable of storing a vast number of individual records and databases for a vast number of brokers/operators, the application may be 'split' or situated on a number of computer servers in order to provide the required load balancing.

Before data is synchronised between the Local Broker Application and the Remote Host Application, initial velocity error checks are performed by the Local Broker Application on each transaction. These velocity error checks are based upon certain rules and parameters maintained at the Local Broker Application (note, some of these rules and parameters may initially have been created by the Remote Host Application and are downloaded to the Local Broker Application). If the transaction passes these initial velocity error checks, the data is then synchronised with the Remote Host Application for further approval and processing.

When a synchronisation is performed, the Remote Host Application firstly identifies, authenticates and validates the credentials of the broker who sent it and exchanges the appropriate security encryption keys before any data is exchanged.

Depending upon the transaction/data type, once received by the Remote Host Application, further velocity error checks may be performed by the Remote Host Application. For example, a new contract is sent to the Remote Host Application which requires a credit check to take place on the borrower. The

Remote Host Application will check to see if an existing credit report exists for this borrower, if not the Remote Host Application will forward an electronic request to a credit ratings agency for a credit report on this borrower. When the credit report and credit score of the borrower is returned, based upon rules and parameters set up on the Remote Host Application, the borrower will be approved or declined for funding.

Other velocity checks which may take place include checking to see if the borrower in question has existing contracts with this broker, or other brokers, and whether a history of rejected or defaulted payment transactions exists for this borrower.

The Remote Host Application includes of a number of master database tables where broker, borrower, quotation, contract and payment information is stored. A number of these tables are created and maintained by the Remote Host Application e.g. the Master Underwriter Database Table and the Master Banks Database Table. Other tables are full replications of the data forwarded by each broker e.g. Master Borrowers Database Table and Master Contracts Database Table.

In order to perform efficient and effective replication of each broker's data, the Remote Host Application uses two methods of functionality to perform the replication process; Relational Theory and Normalisation.

Relational Theory is the term given to the process by which information in each Master Database Table is attached or cross referenced with other records in other Master Database Tables to avoid the unnecessary duplication and re-entry of data.

In order for Relational Theory to take place, each record in a Master Database Table is assigned a primary key (or identification code). In turn, each record within the Master Database Table contains fields which have been assigned 'foreign' keys (identification codes) which identify the Master Database Table they cross reference to.

For example, the Master Contracts Database Table may contain 3 contracts with primary keys of Contract001, Contract002, Contract003. The contract record identified as Contract001 has been created by a broker with a foreign key of BrokerABC and is assigned to a borrower with a foreign key of

BorrowerSmith01. In turn, the Master Borrowers Database Table will contain a record of the borrower with the primary key of BorrowerSmith01 and the Master Brokers Database Table will contain a record of the broker with the primary key of BrokerABC.

Normalisation is the process by which the Remote Host Application receives synchronised data from each broker and breaks the data down into individual pieces in order to be recorded in the appropriate database tables.

As outlined in the Synchronisation section each sequence or string of information which is synchronised between a Local Broker Application and the Remote Host Application contains a 'header' packet of data which is used by the Remote Host Application to identify, authenticate and validate the credentials of the broker who sent it. The header should contain the following data unique identification code of the registered broker organisation, and the unique 'terminal' identification code of the individual local broker application.

In addition, every piece of system data sent in the sequence or string from the Local Broker Application and attached to the header packet of data contains a number of database table and record identification codes, allowing the Remote Host Application to replicate the data in the correct Master database tables recorded at the Host. For example, if a particular contract is forwarded to the Remote Host Application when a synchronisation is performed, the contract data will contain the identification code of the database table it belongs to [TableID], a unique contract ID [ContractID] and the unique ID of the borrower [BorrowerID] that this contract belongs to. Also included in the record are the various data fields attached to the contract such as Loan Amount, Start Date, etc.

Once the sequence or string of information is validated by the Host, the Normalisation function takes place by identifying the primary keys (or identification codes) within the string and replicating the data associated with these primary keys in the appropriate Master Database Tables. For example, the Normalisation function will recognise;

a) the contract's primary key and replicate all the contract information in the Master Contracts Database Table.

b) the borrower's primary key associated with this contract and will replicate the borrower in the Master Borrower's Database Table.

c) the quotation associated with this contract and will replicate the quotation in the Master Quotation's Database Table.

d) the insurance policy associated with this contract and will replicate the policy in the Master Insurance Policy Database Table.

Utilising the process of Relational Theory, the Normalisation function will cross reference the primary keys of the broker, borrower, quotation, underwriter and insurance policy within the Master Contracts Database Table, in this case when recorded under the Master Contracts Database Table, the primary keys of the other fields become 'foreign' keys.

The Remote Host Application performs all necessary settlements between all stakeholders, including monitoring and processing all payments, utilising an electronic interface to an acquiring bank. This process is known as Transaction Automation and is outlined below. It will be understood that the example given is the settlement of a standard contract repayment, however many other settlements may take place such as the payment of commission for the contract to the broker, the payment in full of the premium funded under the contract to the underwriter, the payment of monthly access fees from the broker to the operators of the Remote Host application, etc.

On a daily basis, the Remote Host Application sweeps the Master Scheduled Payments Database Table, detecting all scheduled payments which are to take place on this date. Each scheduled payment is in effect comprised of two payments. One is a debit transaction, removing the funds from the borrower's account, the other is a credit transaction where the funds are deposited into the funder's account.

The Remote Host Application then replicates the necessary information for each payment transaction, creating a new 'batch' of bank transfer records in the Bank Transfer Database Table. At a given time each day, the 'batch' is converted to a sequence or string of data in a format which the acquiring bank accepts. The acquiring bank then performs the necessary debit and credit transactions and returns a sequence or string of data to the Remote Host Application. This data is a DDR File and contains the individual direct debit details of the payments which have taken place, including whether they where performed successfully or not.

The Remote Host Application replicates each direct debit record in the DDR Detail Record Database Table. Based upon whether the bank transfers were completed successfully or not, the Remote Host Application either creates an entry in the Master Rejection Log Database Table or creates an entry in the Master Settlement Database Table. This information is downloaded to the Local Broker Applications when the next synchronisation is performed.

The Remote Host Application ideally includes the primary database tables listed in Figure 1. Many of these database tables are 'master' tables which are replicated on the Local Broker Application. The purpose and functionality of these primary database tables is outlined below in further detail. A flow diagram of the Remote Host Application and its primary database tables is attached as **Figure ?.**

The Master Broker Database Table stores data related to the identity of each broker with whom the Remote Host operator has a 'trading' agreement in place such as name, address, contact details, authorised contact people, company ABN numbers, etc. In addition to general contact information, the Broker Database Table also stores the details of each broker's nominated bank accounts for commission payments, settlement payments, etc.

Each Local Broker Application is assigned a Terminal Identification Code and a Broker Identification Code (calculated from the registration details of the broker) and these codes are stored in the Broker Database Table. These two identification codes are encoded with every record created on the Local Broker Application and are included with every batch of data forwarded to the Remote Host Application when a synchronisation is performed, allowing the Remote Host Application to identify, authenticate, record and process the data from this terminal (application) and operator.

The Remote Host operator has the ability to establish an 'Overall Funding Level' for each funder used by a broker. This can default to 'unlimited' but may vary from broker to broker. The Overall Funding Level prohibits a broker from converting any further contracts over a certain limit and is calculated on the overall amount outstanding at any time.

When a contract is received by the Remote Host Application for approval and processing, the system cross references the total outstanding amount of all contracts which the broker has in place with the Overall Funding Level field recorded in the Master Broker Table for this broker. If the Overall Funding Level will be exceeded by the new contract, the Remote Host Application will automatically return the contract to the Local Broker Application and notify the broker that the overall funding level has been reached.

In addition, should a particular broker be nearing the impending Overall Funding Level established for this broker, the Remote Host Application will automatically send a warning notification to the broker. Alternatively, the Remote Host operator may intervene and increase the overall funding level amount.

The Broker Code Database Table is a summary of the Master Broker Table and stores the Terminal ID code, the Broker Registration Code and the date of the last synchronisation performed by each broker.

The Broker Code Table is used by the Remote Host Application to identify, authenticate and validate each broker when a synchronisation is performed. Each sequence or string of information which is synchronised between a Local Broker Application and the Remote Host Application contains a 'header' packet of data which contains the unique identification code of the registered broker, and the unique 'terminal' identification code of the individual Local Broker Application.

The Remote Host Application cross references this data with the data recorded in the Broker Code Table and either correctly identifies the broker and accepts the synchronisation or declines the data if no match is found, or the data is incorrect.

The Master Preferences Table stores data related to the nominated defaults and preferences for each broker, such as the default commission percentage, minimum and maximum loan term allowed, minimum and maximum loan value allowed, default documentation fees to be charged, etc.

Whilst some of the defaults and preferences are created by the brokers themselves when initially setting up their systems and are forwarded to the Remote Host Application when a synchronisation is performed, other defaults and preferences are set by the Remote Host operator and are downloaded to each Local Broker Application and recorded in the Broker Preferences Table when a synchronisation is performed.

As the invention is capable of handling and processing contract funding from multiple funders through the one application, the Master Funders Table stores data related to each funder with whom the operator of the Remote Host Application has contract funding arrangements in place such as the funding type of each funder (e.g. Insurance Premium Funding, Professional Fee Funding, Membership Fee Funding) and the contact details of each funder.

The invention allows for three methods of processing on behalf of funders and enables each funder to choose one of these methods for processing;

a) Where the broker performs the initial 'introduction and paperwork' on the Local Broker Application, with the Remote Host Application forwarding an electronic settlement file to the funder, who will then perform its own direct debits and settlements.

b) Where the operator of the Remote Host Application performs an end-to-end solution for a funder by performing all the quotation generation, contract generation, settlement and collection on behalf of the funder.

c) Where the broker has the ability to 'self fund' his own book of business. In this case, the broker becomes the funder and utilises his own pool of funds to fund non-approved contracts, up to a set credit limit. The invention performs an end-to-end solution for the broker/funder by performing all the quotation generation, contract generation, settlement and collection on behalf of the broker/funder.

The Funders Table can be closely linked to the master lending rates, master lending rules, cost of funds, banking and fees, brokers performance table and master forms template table sub-set of tables which can contain the funding data and parameters for each funder.

The Master Lending Rate Table stores data related to the margins each funder wishes to make on loans provided to borrowers. The Lending Rate Table is used to calculate the finance charges on all quotations and contracts.

The Master Lending Rate Table is created and maintained by the Remote Host Application. A default margin matrix is applied to every broker upon initial setup on the Remote Host Application. Once configured however, the finance rates may vary from broker to broker depending upon loan size, loan duration, type of funding (i.e. Premium Funding - for either cancellable or non-cancellable business, Professional Fee Funding, Membership Fee Funding) and the frequency of business. Once configured, the funders' finance rates are downloaded to each Local Broker Application and recorded in the Lending Rate Table when a synchronisation is performed.

The Master Lending Rules Table stores data related to each funder's lending rules or general defaults which apply to brokers when performing funding quotations and contracts. Rules are created as a series of 'IF, THAN, AND/OR, NOT, ELSE' statements and apply to features such as credit checking, use of guarantors, contract value, etc.

For example, a lending rule may be created that assigns all contracts under $25,000 to Funder A and all contracts over $25,000 to Funder B. Each funder has an individual set of rules, for example Funder A may require that only one party must sign the contract at the time of acceptance whereas Funder B may require all parties to sign contracts.

These rules are created and maintained by the Remote Host Application and are downloaded to each Local Broker Application and recorded in the Lending Rules Table when a synchronisation is performed. When applicable, modifications to the Lending Rules Table are downloaded during subsequent synchronisations.

The Cost of Funds Database Table stores a matrix of fields related to each funder's lending ability and defaults, such as loan term, loan amount range, loan type, delay in settlement, etc. The Cost of Funds table is used to accurately calculate margins and the overall profitability of each loan from each funder. The Cost of Funds Table is created and maintained by the Remote Host Application.

The Master Banking and Fees Table stores data related to each funder's bank processing charges and fees which apply to brokers when performing funding quotations and contracts such as transaction fees, rejection fees, quotation, contract, documentation and credit checking fees.

When a funder is initially allocated to a broker, a default set of fees and charges is applied to the broker upon initial setup. Once configured by the Remote Host Application however, the fees and charges may vary from broker to broker depending upon loan size, loan duration, type of funding (i.e. Premium

Funding - for either cancellable or non-cancellable business, Professional Fee Funding, Membership Fee Funding) and the frequency of business.

The Banking and Fees Table enables the operator to accurately calculate the cost of processing fees and the profit of a contract, depending on the number of settlements to be made, payments to be processed, rejections for the life of the loan and any credit checks performed.

The Banking and Fees Database Table also stores multiple trust bank account details for each funder, with each funder nominating a default trust account for settlement. Recording the details of multiple trust accounts, enables the Remote Host Operator to override the default trust account and select another when processing a specific transaction which may be outside the normal 'ticket size' of the funder's transactions.

The Broker Performance Database Table stores data related to each broker's 'trading' performance, with each funder.

The Broker Performance Table is represented in a GUI screen, allowing the Remote Host operator to visually display, for each funder, the brokers which have funded with them including individual broker data such as; Total Sold, Total Quoted, Date of Last Quotation, Date of Last Contract, etc. for any given date range.

The Commission Rates Database Table stores data related to the commission rates applicable to each broker. These rates are configured and controlled by the Remote Host Application and may vary from broker to broker. The commission rates may also be modified from time to time by the Remote Host operator.

The commission rate for each broker is forwarded to the Local Broker Application when the broker performs an initial synchronisation, however this information is hidden and is only used by the application when calculating the commission rate for the broker at the time of creating a quotation. When applicable, modifications to the Commission Rates Table are downloaded during subsequent synchronisations.

The Master Underwriter Table is utilised when producing Insurance Premium Funding contracts only and stores data related to insurance underwriters with brokers may do business. The contacts, branch locations, classes of insurance (e.g. cancellable & non- cancellable) and premium bank account details (required when performing premium payments) of every insurance underwriter are maintained by the Remote Host Application and are downloaded to the Local Broker Application and recorded in the Underwriter Database Table when a synchronisation is performed. When applicable, modifications to the Underwriter Table are downloaded during subsequent synchronisations.

When an Insurance Premium Funding contract is finalised and sent to the Remote Host Application for processing, the Remote Host Application identifies the underwriter nominated in the contract (each underwriter in the Master Underwriter Table is assigned a unique ID code) and automatically settles the amount of the insurance policy premium to the underwriter's premium/trust bank account on the broker's behalf.

When the settlement has been performed, the Remote Host Application automatically notifies the underwriter of the settlement payment, electronically. In the event that the borrower defaults on the contract, the Remote Host Application automatically notifies the underwriter to cancel the policy and requests a refund on the pro-rata balance of the contract.

The Master Banks Database Table stores data related to banking and financial institutions with whom brokers and borrowers may do business. The branch name, address and BSB of every banking and financial institution are created and maintained by the Remote Host Application. This table is supplied with the first deployment of each Local Broker Application and maintained thereafter with regular updates.

The Master Credit Card Table stores data related to each credit card provider with whom the Remote Host Application is able to perform settlement with, such as card type and default card charges.

The default credit cards and card charges are maintained by the Remote Host Application and are downloaded to each Local Broker Application and recorded in the Credit Card Table when a synchronisation is performed.

Once downloaded to the Local Broker Application, the broker may individually nominate which credit card providers displayed in the Credit Card

Table the broker will accept, for example, the operator may choose to accept Visa and Mastercard but not Amex or Diners.

The Master Stamp Duty Table stores data related to the appropriate stamp duty fees and charges for each region. This table is maintained by the Remote Host Application and synchronised with the Local Broker Application. The Stamp Duty Table can also be used for the calculation of any applicable state taxes and VAT duty.

The Stamp Duty Table is accessed by the Local Broker Application when the broker creates a quotation for a borrower. Each broker operates within a default region and stamp duty is calculated on the basis of this region. If required, the default region can be overridden at time of creating a quotation allowing another region to be selected in order to calculate the appropriate duty.

The Master Forms Template Table stores the templates, per funder, of all stationery used in the production of quotations, contracts, terms and conditions documentation, guarantor forms, etc.

The application is capable of providing funding on behalf of multiple funders, each of whom has individual contract documentation or will require particular information to be collected from the borrower or to be outlined on the contract documentation.

As such, each stationery template recorded in the Master Forms Template Table is assigned a unique template ID and funder ID, ensuring that the correct contract documentation for each funder is accessed and printed by the system.

In order to maintain the integrity of all stationery templates, all templates are created and maintained by the Remote Host Application. Any new or modified templates are downloaded to the Local Broker Application and replicated in the Forms Template Table on the Local Broker Application when a synchronisation is next performed. If required, an authorised representative of the funder may 'brand' or customise form templates with the broker's logo.

The Master Report Template Table stores the templates of all reports used and/or created in the system. Each Local Broker Application is initially deployed with a number of system reports which are created and maintained by the Remote Host Application. Any new or modified templates are downloaded to the

Local Broker Application and replicated in the Report Template Table on the Local Broker Application when a synchronisation is next performed.

In addition, the broker may create and edit new reports according to the broker's own requirements, which are assigned a name and unique ID and are also recorded in the Report Template Table. Any report created and saved by the broker appears in the list of available reports to select from.

The Email Template Table stores the templates of all emails created by the Remote Host operator and used for automatic notification, such as the Defaulted Payment email template and the Cancellation of Contract email template, etc. Any email template created and saved by the Remote Host operator is assigned a unique ID and description and is recorded in the Email Template List.

Each email template contains a default set of merge codes which access data from all database tables. For example, a Cancellation of Contract email template will include merge codes from the Contracts Table, automatically inserting the details of the cancelled contract, such as the ID number, amount of contract, etc.

The Email Template Table is used in conjunction with the Emails to Send Table in order to perform automatic notification. For example, in the instance that a scheduled payment defaults, the Remote Host Application automatically sends a Defaulted Payment email to the relevant broker and the borrower. The application cross references the Defaulted Payment email template in the Email Template Table with the details of the defaulted payment in the Rejection Log table and automatically creates the notification emails for the broker and borrower. The notification emails are then recorded in the Emails to Send table and forwarded to the broker and borrower.

The Emails to Send Table is used in conjunction with the Email Template Table in order to perform automatic notification. For example, in the instance that a scheduled payment defaults, the Remote Host Application automatically sends a Defaulted Payment email to the relevant broker and the borrower. The application cross references the Defaulted Payment email template in the Email Template Table with the details of the defaulted payment in the Rejection Log table and automatically creates the notification emails for the broker and borrower. The notification emails are then recorded in the Emails to Send table and forwarded to the broker and borrower.

The Master Borrower Table stores data related to every borrower who has a quotation or contract recorded by the Remote Host Application, such as the borrower's contact and address details, employment details, identification details, credit limit, status, etc.

Each borrower is assigned a unique Borrower Identification Code and these codes are stored in the Master Borrower Table. The Borrower Identification Code is recorded with every note, quotation, contract or payment assigned to the borrower.

Borrowers can also be assigned a 'pre-approved' credit limit. This limit is established by the operator of the Remote Host Application and will allow a quotation to take place outside of the normal lending ranges applied to a broker.

The Master Borrower Table also records the details of the borrower's previously stamped stamp duty payments. If the borrower has existing contracts with the broker and an existing funder, the amount of stamp duty already paid to the funder is recorded in the Borrower Table and is automatically displayed on the quotation screen and taken into account when calculating stamp duty for the contract.

If Membership Fee Funding is to be performed using the application, borrowers can be assigned a membership number. A unique 'starting' membership number is created by the Remote Host Application, with the ability to select some or all of the broker's borrowers in order to assign them with a membership number. This 'default' membership number will be automatically displayed when entering a borrower's membership screen, however the broker can override this number and manually allocate a membership number to the borrower.

When a synchronisation is performed between the Local Broker Application and the Remote Host Application, the Local Broker Application automatically detects any new or modified data in the Borrower Database Table and forwards this information (along with the relevant Borrower Identification Codes) to the Remote Host application, where it is replicated in the Master Borrower Table stored at the Host.

Alternatively, should circumstances dictate, the Remote Host operator also has the ability to create a borrower, quotation and contract at the Remote Host Application on a broker's behalf. When a synchronisation is next performed between the Remote Host Application and the broker's Local Broker Application, the newly created borrower/quotation/contract is downloaded and recorded in the broker's local datafiles.

The Master Borrower Notes Table stores the details of every note created in the application which has been assigned to a borrower (identified by the unique Borrower Identification Code and Note Identification Code which are recorded with each note).

Borrower notes can be either 'system notes' which are automatically created by the application (a system entry such as the creation date of a contract) or 'operator notes' which are text notes applicable to the borrower and have been manually entered by the operator (e.g. any pertinent notes related to a client's contract).

In the case of 'system notes' created by the Remote Host application, these are downloaded to the Local Broker Application and recorded in the Borrower Notes Table when the next synchronisation is performed.

In the case of 'operator notes', when a synchronisation is performed between the Local Broker Application and the Remote Host Application, the Local Broker Application automatically detects any new or modified data in the Borrower Notes Table and forwards this information (along with the relevant Borrower Identification Codes) to the Remote Host application, where it is replicated in the Master Borrower Notes Table stored at the Host.

The Borrower Credit History Table stores the details of each borrower's credit report received from a credit reference agency, such as the date the report was created, the date the report was last modified and the ID code of the broker who requested the report.

This information is stored by the Remote Host Application and is used when performing velocity error checks on borrowers at the time of processing a contract. Depending upon the Lending Rules configured for the funder of the contract, the Remote Host Application may be required to perform a credit check on each borrower before approving the contract.

The Remote Host Application will first check the Borrower Credit History Table to see if a credit report already exists for this borrower. If so, the application checks the date of the report to establish if the report is still current and therefore relevant.

If the report is not current or no credit report exists for the borrower in the Borrower Credit History Table, the Remote Host Application refers the borrower's details to a credit reference agency and a credit report file displaying the borrower's credit score is returned and recorded in the Borrower Credit History Table. The application then cross references the borrower's credit score with the parameters established in the Lending Rules for the funder and either approves or declines the contract based upon the borrower's credit score.

The Master Payment Methods Table stores the details of every payment method which has been assigned to every borrower, such as the account name, BSB number, account number, credit card number, etc.

When a synchronisation is performed between the Local Broker Application and the Remote Host Application, the Local Broker Application automatically detects any new or modified data in the Payment Methods Database Table and forwards this information (along with the relevant Borrower Identification Codes) to the Remote Host application, where it is replicated in the Master Payment Methods Table.

The Master Messages Table stores the details of every message created in the application which has been sent to or received from a borrower (identified by the unique Borrower Identification Code and Message Identification Code which is recorded with each message).

The Master Messages Table stores the data for each message such as Message ID, Borrower ID, User ID of the operator who created it, message type (SMS, email, fax, letter) date and time sent/to be sent and message status (either Pending, Sent, Rejected or Received).

When sending SMS, email and fax messages, the Local Broker Application synchronises with the Remote Host Application and forwards the messages to the Host for processing via the Remote Host Application's SMS, email and fax servers. Each message is replicated in full in the Master Messages Table stored on the Remote Host Application. Any messages which are unable to be sent or are returned as unsent are assigned a status of 'Rejected' and are downloaded to the Messages Table on the Local Broker Application when the next synchronisation is performed.

The Master Quotations Table stores the details of every quotation created by each broker such as borrower details, loan start date, number of instalments, draw down dates and amounts, commission, stamp duty, fees and charges and an associated payment schedule for the quotation.

When a synchronisation is performed between the Local Broker Application and the Remote Host Application, the Local Broker Application automatically detects any new or modified data in the Quotations Database Table and forwards this information (along with the relevant Borrower Identification Codes, etc.) to the Remote Host application, where it is replicated in the Master Quotations Table stored at the Host.

Alternatively, should circumstances dictate, the Remote Host operator also has the ability to create a quotation at the Remote Host Application on a broker's behalf. When a synchronisation is next performed between the Remote Host Application and the broker's Local Broker Application, the newly created borrower/quotation/contract is downloaded and recorded in the broker's local datafiles.

The Master Contracts Table stores the details of every contract created by every broker, such as borrower details, loan start date, number of instalments, associated invoice, membership or insurance policy details, commission, stamp duty, fees and charges, an associated payment schedule for the contract and a list of settlements performed during the contract.

The application also allows for 'third party' settlement of contracts. The ability for a broker to perform a third party settlement is configured by the Remote Host Application, and in each case, the borrower must have a valid and current membership number. If a third party settlement is to be performed, the broker selects the third party broker from a list of third party brokers which is configured by the Remote Host Application. The contact and bank account details of the third party broker will then be displayed. The broker then has the option to send the contract to the borrower and third party broker via email, fax or letter for signing and return. Prior to synchronising the contract with the Remote Host

Application, the Local Broker Application will prompt the broker to electronically acknowledge that the borrower and the third party broker have viewed and signed all the appropriate contract documentation such as the contract itself, the terms and conditions and credit checking authorisation forms.

Once a contract has been signed and acknowledged by the broker as executed, a synchronisation is then performed between the Local Broker Application and the Remote Host Application, forwarding the new or modified contract data (along with the relevant Borrower Identification Codes, etc.) to the Remote Host application, where further velocity checks take place according to the Lending Rules established for this funder e.g. the Remote Host application may obtain a credit report on the borrower.

If the contract is accepted by the Remote Host Application after the additional velocity checking takes place a copy of the contract is replicated in the Master Contracts Table stored at the Host. In addition, new entries detailing the contract ID code, borrower ID code, dates, amounts and nominated payment method ID code will be created in the Master Scheduled Payments Table stored at the Host.

Alternatively, at the time of saving a contract, the broker may opt to perform a 'real-time velocity check' of the contract. In this instance the broker must be 'online' (connected to the internet), allowing the Local Broker Application to connect to the Remote Host Application and perform all the necessary velocity error checks on the contract immediately. If approved, the contract moves from a pre-approved state to an approved state. The ability to perform 'real time velocity checking' will be based on rules configured by the Remote Host Application and will be funder dependent, varying from funder to funder and broker to broker. It may also require the system to perform a real-time check with the credit reference agency and thus may take some time.

In addition, should circumstances dictate, the Remote Host operator also has the ability to create a contract at the Remote Host Application on a broker's behalf. When a synchronisation is next performed between the Remote Host Application and the broker's Local Broker Application, the newly created borrower/quotation/contract is downloaded and recorded in the broker's local datafiles.

The Master Guarantors Table stores data related to every guarantor associated with every borrower such as contact and address details, identification details, credit information, etc.

At the time of creating a contract, according to the individual lending rules of each funder (recorded in the Lending Rules Table), it may occur that a borrower's contract may only be approved if a suitable guarantor accepts responsibility for the contract. The details of each guarantor are recorded in the Guarantors Database Table on each Local Broker Application.

When a synchronisation is performed between the Local Broker Application and the Remote Host Application, the Local Broker Application automatically detects any new or modified data in the Guarantors Database Table and forwards this information (along with the relevant Borrower Identification Codes, etc.) to the Remote Host application, where it is replicated in the Master Guarantor Table stored at the Host.

The Master Insurance Policy Table stores the details of every insurance policy attached to a contract in the system and assigned to a borrower (using the unique Borrower Identification Code), such as underwriter, policy number, policy amount, type of policy, etc. When creating an Insurance Premium Funding contract, the broker must assign an insurance policy or policies to the contract.

When a synchronisation is performed between the Local Broker Application and the Remote Host Application, the Local Broker Application automatically detects any new or modified data in the Insurance Policy Table and forwards this information (along with the relevant Borrower and Contract Identification Codes) to the Remote Host application, where it is replicated in the Master Insurance Policy Table stored at the Host.

The Master Invoice Table stores the details of every third party invoice attached to a contract in the system and assigned to a borrower (using the unique Borrower Identification Code), such as invoice date, invoice number, description, etc.

When creating a Professional Fees contract or a Membership Fees contract, the broker must assign an invoice for the professional fees or good and services to be funded to the contract. The broker enters the date, amount and invoice number of the relevant invoice. The broker then selects the Invoice type from a picklist e.g. professional fees, goods and services, both, etc. The data in this picklist is controlled by the Remote Host Application. The broker then enters a description of the professional fees or goods and services covered by the invoice. This information can be selected from a picklist which the broker maintains locally.

When a synchronisation is performed between the Local Broker Application and the Remote Host Application, the Local Broker Application automatically detects any new or modified data in the Invoice Table and forwards this information (along with the relevant Borrower and Contract Identification Codes) to the Remote Host application, where it is replicated in the Master Invoice Table stored at the Host.

The Third Party Broker Database Table stores the details of brokers with whom the Remote Host Application will enable third party settlement of contracts, such as the contact and bank account details of the third party broker.

The ability for a broker to perform a third party settlement is configured by the Remote Host Application. If a broker is authorised to perform third party settlement of contracts, the broker selects the third party broker required from a list of third party brokers on the Contract screen. This list is linked to the Third Party Broker Table which is configured by the Remote Host Application and downloaded to the Local Broker Application.

When the broker has electronically acknowledged that the borrower and the third party broker have viewed and signed all the appropriate contract documentation the contract is synchronised with the Remote Host Application for processing. If approved, the Remote Host Application cross references Third Party Broker Table and uses the bank account details of the nominated third party broker when performing settlement and notification.

The Master Scheduled Payments Table stores the details of every scheduled payment attached to every contract in the system, such as date of instalment, amount of instalment, etc.

When a contract is synchronised with the Remote Host application and approved, the Scheduled Payment Table for this contract is replicated in the Master Scheduled Payments Table stored at the Remote Host.

On a daily basis, the Remote Host Application sweeps the Master Scheduled Payments Database Table, detecting all scheduled payments which are to take place on this date. Each scheduled payment is in effect comprised of two payments. One is a debit transaction, removing the funds from the borrower's account, the other is a credit transaction where the funds are deposited into the funder's account.

The Remote Host Application then replicates the necessary information for each payment transaction, creating a new 'batch' of bank transfer records in the Bank Transfer Table. At a given time each day, the 'batch' is converted to a sequence or string of data in a format which the acquiring bank accepts. The acquiring bank then performs the necessary debit and credit transactions and returns a sequence or string of data to the Remote Host Application. This data is a DDR File and contains the individual direct debit details of the payments which have taken place, including whether they where performed successfully or not.

In the case that a particular scheduled payment is returned as rejected, based upon parameters defined at the Remote Host Application such as the number of times the payment has been rejected, the Host operator will either choose to re-schedule the payment (a suggested re-scheduled payment date will be provided by the Host but can be overridden by the Remote Host operator if not acceptable) in which case the Remote Host Application will insert a 're-scheduled payment' in the Master Scheduled Payments Table upon the rules and parameters defined in the Master Rejection Log Table such as the default date for the next payment attempt (e.g. in 7 days time) and the amount of any additional fees or charges setup in the Master Preferences Table for rejected scheduled payments e.g. $30 for each missed payment. The re-scheduled payment is downloaded to the broker's Schedule Payments Table when a synchronisation is next performed. Alternatively, the Host operator may choose to issue the borrower with a final notice or cancel the contract. The application will then automatically notify the broker and borrower with the appropriate notifications.

The Bank Transfer Table stores the details of every bank transaction performed by the Remote Host Application such as the bank transfer ID, contract ID, borrower ID, broker ID, debit or credit amount, BSB number, account number, etc.

The Bank Transfer Table is used by the Remote Host Application when performing settlement. On a daily basis, the Remote Host Application sweeps the Master Scheduled Payments Database Table, detecting all scheduled payments which are to take place on this date. Each scheduled payment is in effect comprised of two payments. One is a debit transaction, removing the funds from the borrower's account, the other is a credit transaction where the funds are deposited into the funder's account.

The Remote Host Application then replicates the necessary information for each payment transaction, creating a new 'batch' of bank transfer records in the Bank Transfer Table. At a given time each day, the 'batch' is converted to a sequence or string of data in a format which the acquiring bank accepts. The acquiring bank then performs the necessary debit and credit transactions and returns a sequence or string of data to the Remote Host Application. This data is a DDR File and contains the individual direct debit details of the payments which have taken place, including whether they where performed successfully or not.

The DDR Detail Record Table stores the details of every direct debit transaction performed by the acquiring bank such as the batch ID, BSB number, account number, amount, date processed, return code and return description, etc.

The DDR Detail Record Table is used by the Remote Host Application when performing settlement. On a daily basis, the Remote Host Application sweeps the Master Scheduled Payments Database Table, detecting all scheduled payments which are to take place on this date. The Remote Host Application then replicates the necessary information for each payment transaction, creating a new 'batch' of bank transfer records in the Bank Transfer Table. At a given time each day, the 'batch' is converted to a sequence or string of data in a format which the acquiring bank accepts.

The acquiring bank then performs the necessary debit and credit transactions and returns a sequence or string of data to the Remote Host Application. This data is a DDR File and is recorded in the DDR Detail Record Table. Each DDR File contains the individual direct debit details of the payments which have taken place, including whether they where performed successfully or not, as indicated by the return code and return description fields returned with each DDR detail record.

Based upon whether the DDR's were completed successfully or not, the Remote Host Application either creates an entry in the Master Rejection Log Database Table or creates an entry in the Master Settlement Database Table. This information is downloaded to the Local Broker Applications when the next synchronisation is performed.

The Master Settlement Table stores the details of every settlement processed by the Remote Host Application such as settlement status (e.g. pending, paid, etc.) contract ID, payment date, payment amount, bank account details, payee, etc.

To perform settlement, on a daily basis the Remote Host Application sweeps the Master Scheduled Payments Database Table, detecting all scheduled payments which are to take place on this date. The Remote Host Application then replicates the necessary information for each payment transaction, creating a new 'batch' of bank transfer records in the Bank Transfer Table. At a given time each day, the 'batch' is converted to a sequence or string of data in a format which the acquiring bank accepts.

The acquiring bank then performs the necessary debit and credit transactions and returns a sequence or string of data to the Remote Host Application. This data is a DDR File and is recorded in the DDR Detail Record Table. Each DDR File contains the individual direct debit details of the payments which have taken place, including whether they where performed successfully or not, as indicated by the return code and return description fields returned with each DDR detail record.

Based upon whether the DDR's were completed successfully or not, the Remote Host Application either creates an entry in the Master Rejection Log Database Table or creates an entry in the Master Settlement Database Table. This information is downloaded to the Local Broker Application when a synchronisation is performed and replicated in the Settlement Database Table on the Local Broker Application.

The Master Rejection Log Table stores the details of every scheduled payment which has been returned as rejected by the Remote Host Application, such as the borrower ID code, contract ID code, bank transfer ID, retry days, number of rejections for this payment etc.

if a particular scheduled payment is returned as rejected, a record is created in the Master Rejection Log Table and the application will create a 're-scheduled payment' in the Master Scheduled Payments Table upon the rules and parameters defined in the Rejection Log Table such as the default date for the next payment attempt (e.g. in 7 days time) and the amount of any additional fees or charges setup in the Master Preferences Table for rejected scheduled payments e.g. $30 for each missed payment. The modified data is uploaded to the Master Scheduled Payments Table stored at the Remote Host when a synchronisation is next performed.

The Rejection Log Table keeps a record of the number of rejection attempts for each scheduled payment (this information is also cross referenced in the Master Scheduled Payments Table) and after a predefined number of attempts which can be stored in the Master Preferences Table, will forward the details of the outstanding amount to the operator or nominated debt collector for payment.

This information is sourced and maintained by the Remote Host Application and is downloaded to the Local Broker Application when a synchronisation is performed and replicated in the Rejection Log Table on the Local Broker Application.

The Master Audit Trail Table tracks the creation and progress of every quotation, contract and borrower in the system, recording the user name, times and dates of every function associated with each record.

For example, when each broker prints a quotation or answers each Yes/No question when finalising a contract, the user name, date and time are recorded in an Audit Trail Table on each Local Broker Application. If these functions are performed more than once, each occasion is recorded.

When a synchronisation is performed between the Local Broker Application and the Remote Host Application, the data from the Audit Trail Table on the Local Broker Application is uploaded and recorded in the Master Audit Trail Table stored on the Remote Host Application.

The Master Last Synchronisation Table stores a summary of the details of the very last synchronisation performed between each Local Broker Application and the Remote Host Application such as the date and time the synchronisation was performed.

The Master Last Synchronisation Table is used by the Remote Host Application for cross referencing and backup purposes. For example, should the data on a Local Broker Application become corrupted and if the broker has a recent backup copy of the data, the broker can restore the latest backup. If work has been performed and synchronised on the application since the backup was made, the broker can request a copy of the data from the very last synchronisation (this information is recorded in the Master Last Synchronisation Table).

If the broker does not have a recent backup copy of the data, the broker can request a 'Restore of All Data from the Host' which will include all the broker's information which has been replicated at the Host, up to the very last synchronisation performed.

The Master Synchronisation Conflicts Table stores the details of any errors, conflicts or failed synchronisations which have occurred during past synchronisations, such as the date and time the synchronisation conflicts occurred and the database table and fields which were affected.

This information is automatically created and maintained by the Remote Host Application. Should a synchronisation conflict occur, the data is downloaded from the Master Synchronisation Error Table to the Local Broker Application and replicated in the Synchronisation Conflicts Table.

If a problem is severe then the system should prompt the broker to contact the Remote Host operator for assistance. The Remote Host operator will instruct the broker on any processes which need to be performed in order to correct the conflict.

The Web Interface Application is a 'cut down' Internet based version of the Local Broker Application which enables brokers to remotely access their own datafiles and create prospects, borrowers, quotations, contracts and messages (SMS, email, and fax) via the Internet from any computer or location via a web browser.

The Remote Host operator 'publishes' the GUI [graphical user interface] screens of the Web Interface Application and a subset of the broker's database tables (taken from the master database tables stored on the Remote Host Application) onto the web via a HTML page generator allowing the broker to enter, view, select and report on the stored data.

All transactions performed on the Web Interface Application are duplicated and recorded at the Remote Host Application. Upon the next synchronisation between the Remote Host Application and the Local Broker Application, the transactions performed on the Web Interface Application are synchronised to the Local Broker Application in order to update the broker's local datafiles.

It will be understood that the system, in its present form, can be adapted to suit other financial services, industries and products such as Residential Mortgages and Receivables Funding, as required.

Traditional contract funding systems utilise either costly, time consuming manual systems, or antiquated computerised systems with limited flexibility in document generation, payment methods and reporting options. These problems are overcome by the Invention. The invention is a new business process which electronically automates the entire contract funding process using unique 'store and forward' technology. The invention effectively removes the manual management and processing of an operator's contract funding business.

DATED this 6th day of October 2003

NAPOLEON CORPORATION AS TRUSTEE FOR THE TECHNOLOGY TRUST

WATERMARK PATENT & TRADE MARK ATTORNEYS
GPO BOX 2512
PERTH WA 6001
AUSTRALIA

P23030AUP1 PVF/KMJ

FLOW DIAGRAM 1:

Overall Schematic of Local Broker Application and Connection to Host Application

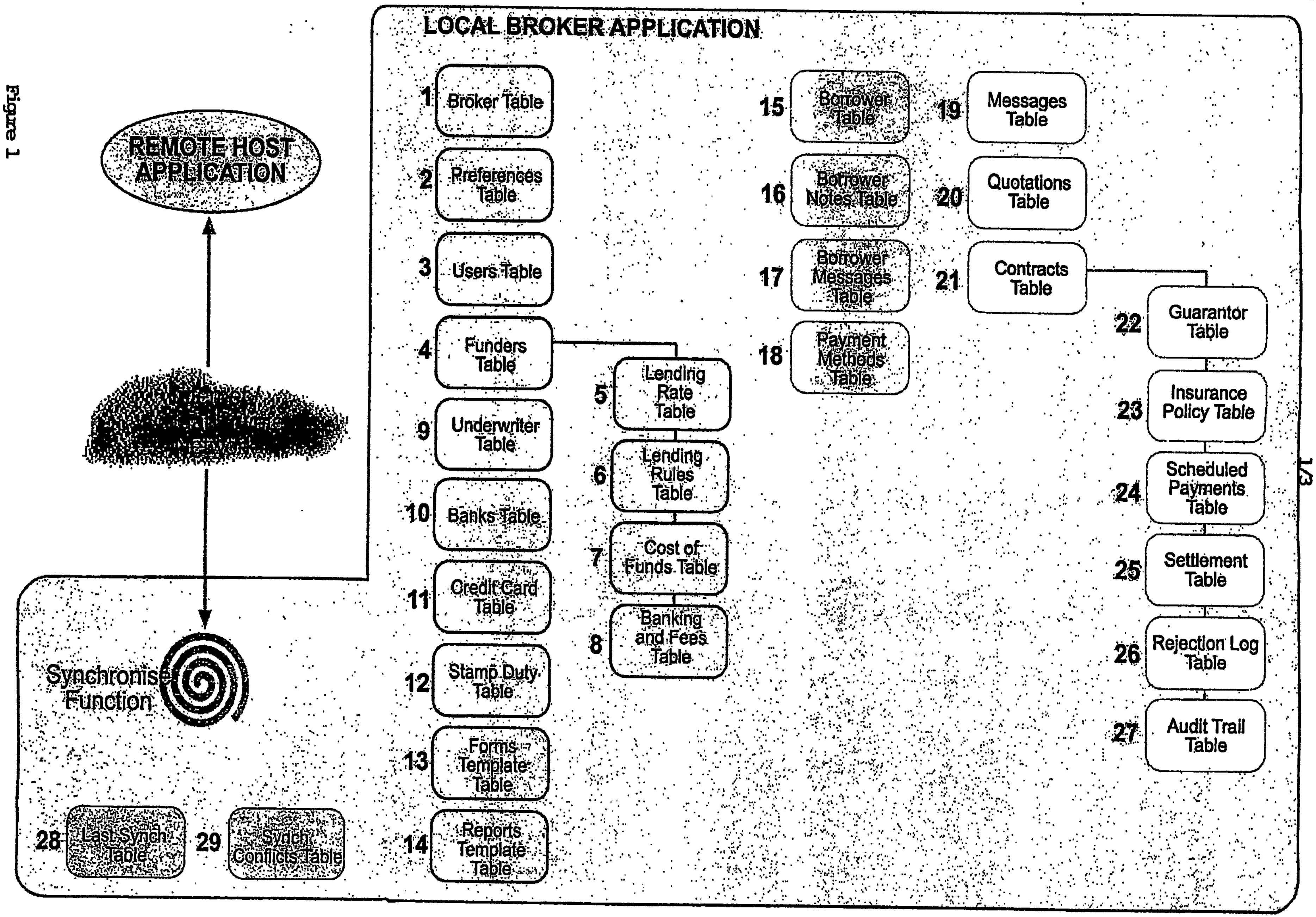


Figure 1

FLOW DIAGRAM 2:

Overall Schematic of Remote Host Application and Connection to Local Broker Application

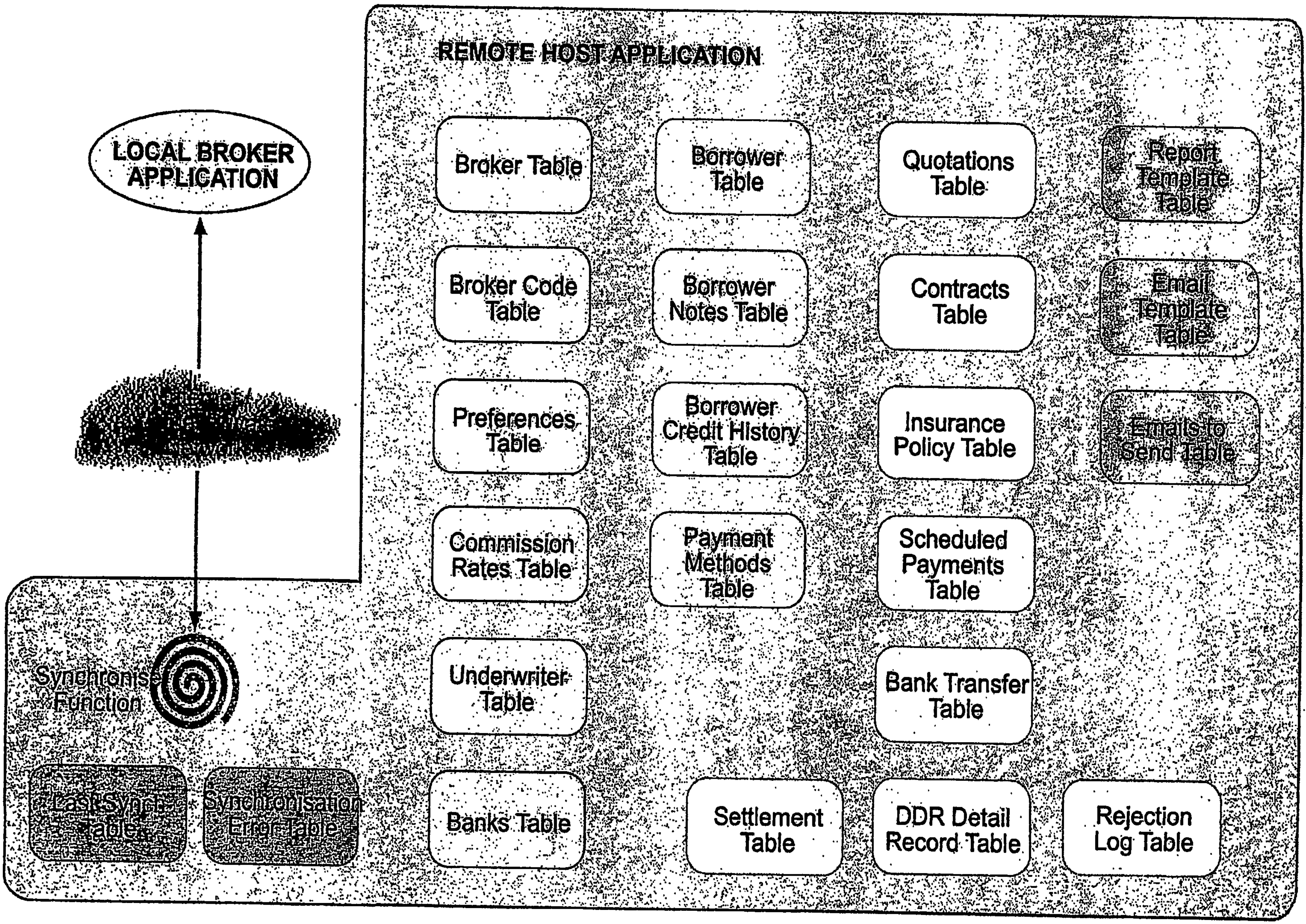
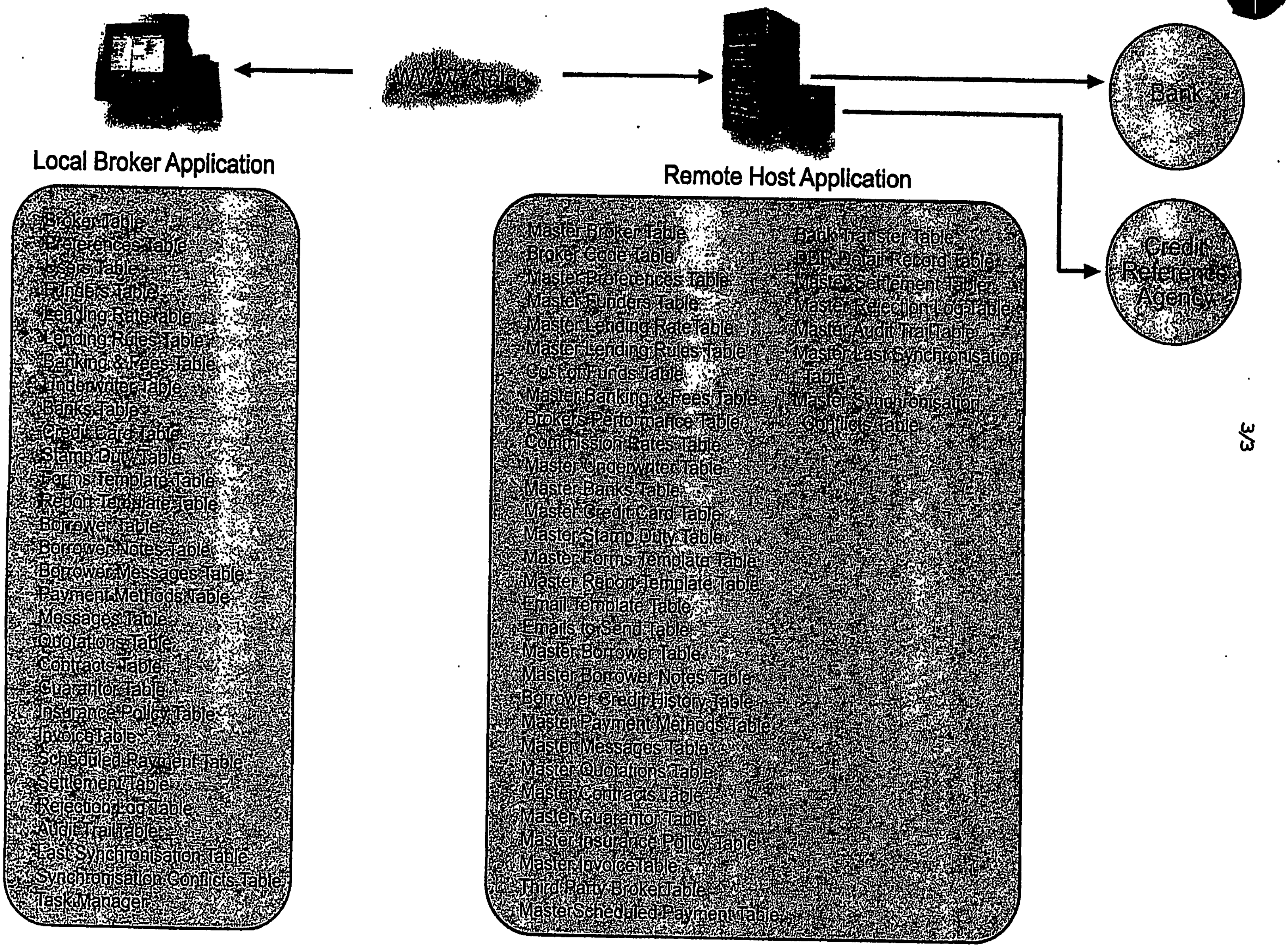


Figure 2

Automated Premium Funding System Schematic



Local Broker Application

Broker Table
Preferences Table
Users Table
Funders Table
Lending Rate Table
Lending Rules Table
Banking & Fees Table
Underwriter Table
Banks Table
Credit Card Table
Stamp Duty Table
Forms Template Table
Report Template Table
Borrower Table
Borrower Notes Table
Borrower Messages Table
Payment Methods Table
Messages Table
Quotations Table
Contacts Table
Guarantor Table
Insurance Policy Table
Invoice Table
Scheduled Payment Table
Settlement Table
Rejection Log Table
Audit Trail Table
Last Synchronisation Table
Synchronisation Conflicts Table
Task Manager

Remote Host Application

Master Broker Table
Broker Code Table
Master Preferences Table
Master Funders Table
Master Lending Rate Table
Master Lending Rules Table
Cost of Funds Table
Master Banking & Fees Table
Brokers Performance Table
Commission Rates Table
Master Underwriter Table
Master Banks Table
Master Credit Card Table
Master Stamp Duty Table
Master Forms Template Table
Master Report Template Table
Email Template Table
Emails to Send Table
Master Borrower Table
Master Borrower Notes Table
Borrower Credit History Table
Master Payment Methods Table
Master Messages Table
Master Quotations Table
Master Contracts Table
Master Guarantor Table
Master Insurance Policy Table
Master Invoice Table
Third Party Broker Table
Master Scheduled Payment Table

Bank Transfer Table
DDR Detail Record Table
Master Settlement Table
Master Rejection Log Table
Master Audit Trail Table
Master Last Synchronisation Table
Master Synchronisation Conflicts Table

Figure 3

Document made available under the Patent Cooperation Treaty (PCT)

International application number: PCT/AU04/001355

International filing date: 06 October 2004 (06.10.2004)

Document type: Certified copy of priority document

Document details: Country/Office: AU
Number: 2003905438
Filing date: 06 October 2003 (06.10.2003)

Date of receipt at the International Bureau: 01 November 2004 (01.11.2004)

Remark: Priority document submitted or transmitted to the International Bureau in compliance with Rule 17.1(a) or (b)

World Intellectual Property Organization (WIPO) - Geneva, Switzerland
Organisation Mondiale de la Propriété Intellectuelle (OMPI) - Genève, Suisse

This Page is Inserted by IFW Indexing and Scanning Operations and is not part of the Official Record

BEST AVAILABLE IMAGES

Defective images within this document are accurate representations of the original documents submitted by the applicant.

Defects in the images include but are not limited to the items checked:

- □ BLACK BORDERS
- □ IMAGE CUT OFF AT TOP, BOTTOM OR SIDES
- □ FADED TEXT OR DRAWING
- □ BLURRED OR ILLEGIBLE TEXT OR DRAWING
- □ SKEWED/SLANTED IMAGES
- □ COLOR OR BLACK AND WHITE PHOTOGRAPHS
- □ GRAY SCALE DOCUMENTS
- □ LINES OR MARKS ON ORIGINAL DOCUMENT
- ☑ REFERENCE(S) OR EXHIBIT(S) SUBMITTED ARE POOR QUALITY
- □ OTHER: ______________________

IMAGES ARE BEST AVAILABLE COPY.
As rescanning these documents will not correct the image problems checked, please do not report these problems to the IFW Image Problem Mailbox.